

**Partial English Translation of Japanese Patent Application  
Laid-Open No.H06-103186**

(54) [Title of the Invention] Gateway System

5 (57) [Abstract]

[Purpose] To effectively manage the connection between a gateway and various devices on a network by various protocols.

[Constitution]

10 A gateway system 10 which is connected to a common transmission medium 13 to which a plurality of host computers 11<sub>1</sub>-11<sub>n</sub>, printers 12<sub>1</sub>-12<sub>m</sub> and other devices having different communication protocols are connected, and executes data transfer among these plurality of devices  
15 while converting protocols, wherein a management means 18 for managing each connection with these devices by a connection identifier and connection type is disposed.

[0008]

20 [Effect]

According to the present invention (Claim 1), the first management means manages the connection by the connection identifier and connection type. The connection identifier is added for each type (kind) of connection to a host name  
25 and spool name pair in the case of an FTP client, and to a pair of host name and user name in the case of FTP server functions and XNS server functions. In this example, the

above mentioned connection identifier is determined by a host name and user name pair when connection is established, an access check is performed, and information on connection thereafter is acquired based on the connection identifier.

5 In this way, connection is managed based on the connection identifier which is determined for each type of connection, so connection can be managed efficiently in a gateway which handles a plurality of protocols.

[0009]

10 Also according to the present invention (Claim 2), the second management means delays the release of connection when the next processing request is received within a predetermined time after connection is established. Therefore when data is frequently exchanged, data can be  
15 exchanged repeatedly without ending an already established connection, and data can be exchanged efficiently.

[0010]

Also according to the present invention (Claim 3), means of accepting a print request by a communication  
20 protocol supported by the printer (XNS server functions) is disposed, so even a host which was once directly requests printing to a printer can now use the gateway, where connectivity improves. Also by accepting a print request via the gateway in this way, acceptance of a print request  
25 is not restricted by the status at the printer side, such as a restriction of the spool area of the printer, restriction of connections which can be connected to the printer

simultaneously, and non-acceptance of a request when a printer is down.

[0011]

[Embodiments]

5 (First Embodiment)

Fig. 1 shows an outline of the configuration of the first example of the present invention. A plurality of host computers (hereafter "hosts")  $11_1-11_n$  having FTP (File Transfer Protocol) protocol or XNS (Xerox Network System) protocol, a plurality of printers  $12_1-12_n$  having XNS protocol, and a gateway 10 having both FTP and XNS protocols exist on the same logical network 13. The gateway 10 has three ways to receive as a file reception processing section, that is, an FTP client section 14, an FTP server section 15, and an XNS server section 16, and has an XNS printing client section 17 as a file transmission processing section. The gateway 10 is also provided with a spool 19 for temporarily storing a received file.

[0012]

20 File transfer will first be briefly explained. The following are the three types of connections for a file transfer from a host to the gateway.

(1) Connection by the FTP client section 14 where the gateway acquires a file on the host explicitly using the TCP communication protocol.

25

(2) Connection by the FTP server section 15 where the gateway receives a file sent from the host using TCP communication protocol.

(3) Connection by the XNS server section 16 where the gateway receives a file sent from the host using the XNS communication protocol.

[0013]

Fig. 2 shows the data flow [1], [2] and [3] by connection (1). Logically the host, printers and the gateway exist on the same network system. Fig. 2 shows the file transmission from the host to the printer on this network system.

[1] The gateway (FTP client section) connects to the host by FTP (File Transfer Protocol), and regularly checks whether a print request file exists using the NLST command of FTP.

[2] If a print request file exists on the host, the gateway issues the RETR command of FTP, and directly receives the file by the spool directory of the gateway.

[3] The gateway (XNS printing client section) transmits the file in the spool directory to the printer by the XNS printing protocol.

[0014]

Fig. 3 shows the data flow [1] and [2] in the above mentioned connection (2) or (3).

[1] The FTP host (or XNS host) transmits a file to the gateway.

[2] The gateway transmits a file in the spool directory to the printer by XNS printing (printing protocol).

US

(19) 日本国特許庁 (JP)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開平 6 - 1 0 3 1 8 6

(43) 公開日 平成 6 年 (1994) 4 月 15 日

(51) Int. Cl. <sup>5</sup> 識別記号 庁内整理番号 F I 技術表示箇所  
G 0 6 F 13/00 3 5 1 B 7368 - 5 B  
H 0 4 L 29/06  
8220 - 5 K H 0 4 L 13/00 3 0 5 B

審査請求 未請求 請求項の数 3

(全 3 0 頁)

(21) 出願番号 特願平 4 - 276736

(22) 出願日 平成 4 年 (1992) 9 月 22 日

(71) 出願人 000005496

富士ゼロックス株式会社  
東京都港区赤坂三丁目 3 番 5 号

(72) 発明者 市川 正人

神奈川県川崎市高津区坂戸 3 丁目 2 番 1 号  
K S P / R & D ビジネスパークビル 富士  
ゼロックス株式会社内

(72) 発明者 中澤 俊哉

神奈川県川崎市高津区坂戸 3 丁目 2 番 1 号  
K S P / R & D ビジネスパークビル 富士  
ゼロックス株式会社内

(74) 代理人 弁理士 岩上 昇一 (外 3 名)

最終頁に続く

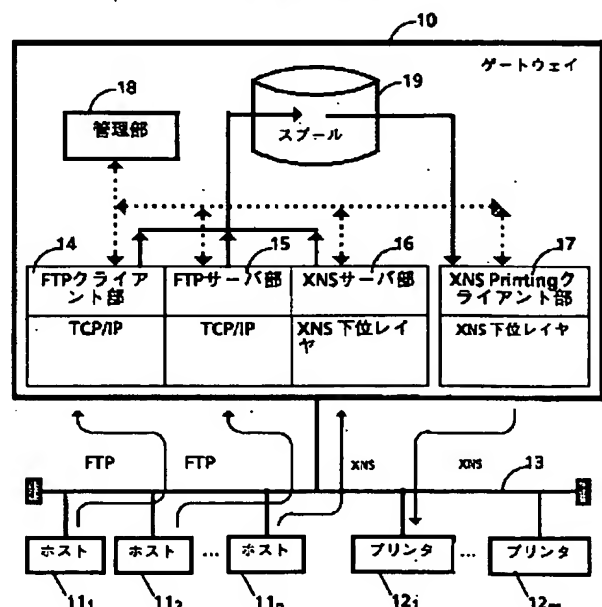
(54) 【発明の名称】 ゲートウェイシステム

(57) 【要約】

【目的】 ゲートウェイとネットワーク上の諸装置との間の種々のプロトコルによるコネクションを効率的に管理すること。

【構成】 異なる通信プロトコルを持つ複数のホストコンピュータ 11<sub>1</sub> ~ 11<sub>n</sub> やプリンタ 12<sub>1</sub> ~ 12<sub>m</sub> 等の装置が接続される共通の伝送媒体 13 に対し接続され、それらの複数の装置間でプロトコル変換可能にデータ転送するゲートウェイシステム 10 において、前記装置に対する各コネクションをコネクション識別子とコネクションタイプで管理する管理手段 18 を設けた。

図 1 第 1 の実施例



## 【特許請求の範囲】

【請求項1】 異なる通信プロトコルを持つ複数の装置が接続される共通の伝送媒体に対し単一のインタフェースで接続され、それらの複数の装置間でプロトコル変換可能にデータ転送するゲートウェイシステムにおいて、前記装置に対する各コネクションについてコネクション識別子とコネクションタイプで管理する第1の管理手段を設けたことを特徴とするゲートウェイシステム。

【請求項2】 コネクションが確立された後、所定時間内に次の処理要求があるときにはコネクションの解放を遅延させる第2の管理手段を設けたことを特徴とする請求項1記載のゲートウェイシステム。

【請求項3】 前記異なる通信プロトコルを持つ複数の装置には、ある通信プロトコルをサポートするプリンタと、そのプリンタのサポートする通信プロトコルでプリント要求を行うことのできるデータ処理装置が含まれており、かつ、ゲートウェイシステムには前記プリンタのサポートする通信プロトコルによるプリント要求を受け付け、その受け付けた要求を前記プリンタのサポートする通信プロトコルによりプリンタへ送信する手段を設けたことを特徴とする請求項1記載のゲートウェイシステム。

## 【発明の詳細な説明】

## 【0001】

【産業上の利用分野】本発明は、異なる通信プロトコルを持つ複数の装置間でプロトコル変換可能にデータ転送するゲートウェイシステムに関する。

## 【0002】

【従来の技術】従来から、異なる通信プロトコルを持つデータ処理装置（ホストコンピュータ）とプリンタが同一の伝送媒体上に接続されたネットワークシステムにおいて、異なるプロトコル間でのプロトコル変換を行うためのマルチプロトコル、マルチホスト、マルチプリンタをサポートするゲートウェイを設けるシステムは種々提案されており、ホストコンピュータ（以下、単に「ホスト」という）からゲートウェイへのファイルの転送方法としては、ゲートウェイがホスト上のファイルを獲得しに行くFTP（File Transfer Protocol：ファイル転送プロトコル）クライアント機能によるもの（例えば、特開平4-155558号公報）、FTPクライアント機能の他にさらにホストからの接続要求を受け付けてファイルを受信するFTPサーバ機能をも持つもの（例えば、特開平3-75949号公報）などがある。また、異なる通信プロトコルを持つホストからそれぞれのプロトコルでプリンタ制御装置へプリント要求ファイルを送り、プリンタ制御装置でプロトコル変換を行う技術も知られている（例えば、特開昭63-47174号公報）

データ処理装置を他のデータ処理装置と通信制御装置を介してコネクションを確立する際に、コネクションを一

意に識別するコネクション識別子を設定することによりコネクションの管理を行う技術もある（例えば、特開平3-113660号公報）。

## 【0003】

【発明が解決しようとする課題】従来の前記FTPクライアント機能のみによりゲートウェイがホスト上のファイルを獲得するシステム（特開平4-155558号公報）においては、ホスト側からコネクションの開設要求を受け付けることができないという不便があった。なお、そのコネクションはホスト識別子とスプール識別子の2つで決定され、管理されていた。一方、前記FTPクライアント機能と記FTPサーバ機能の両方を備えたシステム（特開平3-75949号公報）においては、通信プロトコルFTPのホストからのコネクション開設要求をも受け付けることができるようになって接続性が向上しているが、コネクションの管理に前記のホスト識別子やスプール識別子の他にコネクションタイプをも考慮する必要があるため、コネクション管理が複雑となっていた。また、これらの従来システムにおいて、ゲートウェイとプリンタ間で要求/データの交換をする場合、まずOSI 7階層の下位レイヤ（ネットワーク層）でコネクション（通信路）を開設し、そのコネクションを利用して上位レイヤのデータを交換し、データの交換が終了したら必ずコネクションを解放していた。この方式だと、頻繁にデータの交換が行われる場合、そのたびに必ずコネクションの開設/解放を行なわなければならなく効率のよいデータ交換ができなかった。また、上に述べたようなゲートウェイを設けずにホストから通信プロトコルXNS等のプリンタのサポートする通信プロトコルで直接にプリンタに送信することは、従来、広範に実施されているが、プリンタのスプール領域の制限、プリンタに同時に接続できるコネクション数の制約、プリンタダウン時はプリント要求が全くできなかった等のプリンタ側の状態にプリント要求受付が左右されることがあった。

【0004】本発明は、上記の問題を解決することを課題とするものである。即ち、本発明はゲートウェイとネットワーク上の諸装置との間の種々のプロトコルによるコネクションを効率的に管理することのできるゲートウェイシステムを得ることを目的とするものである。また、本発明は頻繁にデータの交換が行われる場合、より交換の効率を向上させることのできるゲートウェイシステムを得ることを目的とするものである。また、本発明はプリンタのサポートする通信プロトコルでホストからプリントファイルを送信する場合でも、プリンタ側の状態に左右されずに要求受付が可能なゲートウェイシステムを得ることを目的とするものである。

## 【0005】

【課題を解決するための手段】本発明（請求項1）は、異なる通信プロトコルを持つ複数の装置（図1の11、

～11<sub>n</sub>、12<sub>1</sub>～12<sub>m</sub>が接続される共通の伝送媒体  
(図1の13)に対し単一のインタフェースで接続され、それらの複数の装置間でプロトコル変換可能にデータ転送するゲートウェイシステム(図1の10)において、前記装置に対する各コネクションについてコネクション識別子とコネクションタイプで管理する第1の管理手段(図1の18)を設けたゲートウェイシステムである。

【0006】また、本発明(請求項2)は、前記ゲートウェイシステムにおいて、前記コネクションが確立された後、所定時間内に次の処理要求があるときにはコネクションの解放を遅延させる第2の管理手段(図26の267、268)を設けた構成とすることができる。

【0007】また、本発明(請求項3)は、前記ゲートウェイシステムにおいて、前記異なる通信プロトコルを持つ複数の装置(図1の11<sub>1</sub>～11<sub>n</sub>、12<sub>1</sub>～12<sub>m</sub>)には、ある通信プロトコル(XNS)をサポートするプリンタ(図1の12<sub>1</sub>)と、そのプリンタのサポートする通信プロトコルでプリント要求を行うことのできるデータ処理装置(図1の11<sub>n</sub>)が含まれており、かつ、ゲートウェイシステムは前記プリンタのサポートする通信プロトコルによるプリント要求を受け付ける手段(図1の15)と、その受け付けた要求を前記プリンタのサポートする通信プロトコルによりプリンタへ送信する手段(図1の17)とを設けた構成とすることができる。

【0008】

【作用】本発明(請求項1)によれば、第1の管理手段は、コネクション識別子とコネクションタイプでコネクションの管理をする。コネクション識別子は、例えば、FTPクライアント機能の場合にはホスト名とスプール名の対に対して、FTPサーバ機能やXNSサーバ機能の場合にはホスト名とユーザ名の対に対してコネクションのタイプ(種類)ごとに付与されている。また、この例においてコネクション確立時にホスト名とユーザ名の対から上記のコネクション識別子を求め、アクセスチェックを行い、以降はコネクションに関する情報は、コネクション識別子を基に獲得する。このように、コネクションの管理をコネクションのタイプ毎に定められるコネクション識別子を基に行うので、複数のプロトコルを取り扱うゲートウェイにおいてコネクションの管理を効率的に行うことができる。

【0009】また、本発明(請求項2)によれば、第2の管理手段は、コネクションが確立された後、所定時間内に次の処理要求があるときにはコネクションの解放を遅延させる。従って、頻繁にデータ交換が行われる場合には一度開設したコネクションを終わらせることなく何度も利用することができ、効率よくデータの交換を行うことができる。

【0010】また、本発明(請求項3)によれば、プリンタのサポートする通信プロトコルによるプリント要求

を受け付ける手段(XNSサーバ機能)を設けたので、プリンタに対し直接にプリント要求をしていたホストからもゲートウェイを利用することができるようになり、接続性が向上した。また、このようにプリント要求をゲートウェイを介して受け付けるようにしたことにより、プリンタのスプール領域の制限、プリンタに同時に接続できるコネクションの制約、プリンタダウン時の受付不能等のプリンタ側の状態にプリント要求受け付けが制限されることがなくなった。

【0011】

【実施例】

(第1の実施例) 図1は本発明の第1の実施例の構成の概略を示すものである。プロトコルFTP(File Transfer Protocol)またはプロトコルXNS(Xerox Network System)を有するホストコンピュータ(以下、単に「ホスト」と呼ぶ)が複数台11<sub>1</sub>～11<sub>n</sub>、プロトコルXNSを有するプリンタが複数台12<sub>1</sub>～12<sub>m</sub>、およびプロトコルFTP、プロトコルXNSの両方を有するゲートウェイ10が同一の論理ネットワーク13上に存在して構成されている。ゲートウェイ10はファイルの受信処理部としてFTPクライアント部14、FTPサーバ部15、XNSサーバ部16の3つの受け口を有し、ファイルの送信処理部としてXNS Printingクライアント部17を有する。また、受信したファイルを一時的に格納するためのスプール19がある。

【0012】まず、ファイルの転送の概略を説明する。ホストからゲートウェイへのファイル転送のためのコネクションには以下の3通りのタイプがある。

(1) 通信プロトコルTCPを用いてゲートウェイが明示的にホスト上のファイルを獲得しに行くFTPクライアント部14によるコネクション。

(2) 通信プロトコルTCPを用いてホストから送られてくるファイルを受信するFTPサーバ部15によるコネクション。

(3) 通信プロトコルXNSを用いてホストから送られてくるファイルを受信するXNSサーバ部16によるコネクション。

【0013】図2は(1)番目のコネクションでのデータの流れ①②③を示す。論理的に同一のネットワークシステム上にホスト、プリンタ、ゲートウェイが存在している。このネットワークシステム上でのホストからプリンタへのファイル送信を示す。

① ゲートウェイ(FTPクライアント部)はホストへFTP(ファイル転送プロトコル)で接続し定期的にプリント要求ファイルがあるかどうかFTPのNLSTコマンドでチェックしに行く。

② ホストにプリント要求ファイルがあればFTPのRETRコマンドを発行しファイルをゲートウェイのスプールディレクトリへ受信する。



③ ゲートウェイ (XNS Printingクライアント部) はスプールディレクトリにあるファイルをXNS Printingプロトコルでプリンタへ送信する。

【0014】図3は前記(2)または(3)の接続でのデータの流れ①②をに示す。

① FTPホスト (またはXNSホスト) はゲートウェイに対してファイルを送信する。

② ゲートウェイはスプールディレクトリにあるファイルをXNS Printing (プリンティングプロトコル) でプリンタへ送信する。

【0015】図4は本実施例におけるゲートウェイの接続に関する機能を示す図である。即ち、このゲートウェイは図4に示すように、ネットワーク上で接続するホストおよびプリンタの管理を行うための接続情報を格納する接続情報ファイル記憶部41と、接続情報ファイル記憶部41に記憶された接続情報ファイルの内容に基づいて接続の管理を行う管理部42と、各接続タイプごとに接続受信処理を行うFTP接続受信処理部43、XNS接続受信処理部44、スプール接続受信処理部45とを含んでいる。接続情報ファイル記憶部41は、スプール接続情報ファイル411、FTP接続情報ファイル412、XNS接続情報ファイル413およびプリンタ接続情報ファイル414を記憶している。管理部42は、接続情報ファイル記憶部41の各接続情報ファイルを、初期化時にメモリ422に読み込むファイル読み込み部421と、情報を記憶するメモリ422と、メモリ422の内容を読み出すためのメモリ読み出し部423と、アクセスチェック部424と、情報獲得部425を有する。FTP接続受信処理部43、XNS接続受信処理部44、およびスプール接続受信処理部45は、それぞれ図1のFTPサーバ部15、XNSサーバ部16、およびFTPクライアント部14の機能に対応している。

【0016】接続情報ファイル記憶部41の前記4つのファイルはそれぞれ以下に示すような情報を格納するものである。

(1) スプール接続情報ファイル411  
スプール接続情報ファイル411はスプール接続 (FTPクライアント機能) で使用するホストの情報を記述したものである。図5はこのスプール接続情報ファイル411の記述形式の例および記述例を示すものである。この記述例では、スプール接続情報ファイル411に/etc/gw/hostsというファイル名が付けられ、2つのホストが登録されている。ひとつめのエントリはホスト名がacos77、ホストのタイプがACOS6、そのホストへのロ

グインする時のユーザ名がnakazawa、パスワードがnakazawa、アカウントがtoshi、ファイルのストアされているディレクトリ名が/FXSPOOL、ファイル名情報名としてfilename1を使用、プリンタアサイン情報名としてprinter\_assign2を使用することを示す。ファイル名情報はファイル名情報ファイル、プリンタアサイン情報はプリンタアサイン情報ファイルに格納されており、これらについては後ほど説明する。

【0017】(2) FTP接続情報ファイル412

FTP接続情報ファイル412は、FTP接続 (FTPサーバ機能) を使用するホスト：ユーザを記述したものである。図6はこのFTP接続情報ファイル412の記述形式の例および記述例を示すものである。この記述例では、FTP接続情報ファイル412に/etc/gw/ftphostsというファイル名が付けられ、「ホスト+ユーザ」単位で3組のものが登録されている。ひとつめのエントリは「ホスト名+ユーザ名」が「chariot+nakazawa」で、ファイル名情報としてfilename1を使用、プリンタアサイン情報名としてprinter\_assign1を使用することを示す。

【0018】(3) XNS接続情報ファイル413

XNS接続情報ファイル413は、XNS接続 (XNSサーバ機能) を使用するホスト：ユーザを記述したものである。図7はこのXNS接続情報ファイル413の記述形式の例および記述例を示すものである。この記述例では、XNS接続情報ファイル413に/etc/gw/xnshostsというファイル名が付けられ、「ホスト+ユーザ」単位で2組のものが登録されている。ひとつめのエントリはアドレスが00001cf7:080020001cf7、「ホスト名+ユーザ名」が「VAX780@1+vaxtar0」で、ファイル名情報としてfilename1を使用、プリンタアサイン情報名としてprinter\_assign1を使用することを示す。

【0019】(4) プリンタ接続情報ファイル414

プリンタ接続情報ファイル414は、プリンタ接続 (XNSクライアント機能) におけるプリンタの情報を記述したものである。図8はプリンタ接続情報ファイル414の記述形式の例および記述例を示すものである。このファイルの記述例では、プリンタ接続情報ファイル414に/etc/gw/printerというファイル名が付けられ、4組のプリンタが登録されている。ひとつめのエントリはプリンタ名が67901ps1で、プリンタのタイプが6790LPS、アドレスが00001cf7:08003

7002939であることを示す。

【0020】前述のファイル名情報ファイルおよびプリンタアサイン情報名ファイルは次のような機能を有している。

a) ファイル名情報ファイル

ファイル名の何文字目（ファイル名の先頭から何文字目まで）が、プリンタをグルーピングするためのクラス名として使用されるかを、次に例示するような形式で、ファイル名情報ファイルに記述する。

filename1:1:2-7

filename2:1-2:3-7

この例では、filename1はファイル名の1文字目をクラス情報として使用し2-7文字目をファイル受信順序の文字列として使用することを意味する。同様にfilename2は1-2文字目までクラス情報として使用し3-7文字目をファイル受信順序の文字列として使用することを意味する。

【0021】b) プリンタアサイン情報名ファイル

ここでは、67901ps1, 67901ps2, 67901ps7, 67901ps9という名前の4台のプリンタがあり、プリンタをグルーピングするためにクラスという概念を用意する。クラス名とプリンタ名の対応（＝プリンタアサイン情報）をプリンタアサイン情報ファイルに記述する。図9にクラス定義ファイルの一例を示す。ファイル名情報とプリンタアサイン情報は、各サーバ/クライアント機能において、「ホスト+ユーザ」または「ホスト+ディレクトリ」の単位で設定できる。図6のFTPサーバ機能における接続ホストを記述したFTPコネクション情報ファイルの例では、最初のエントリの「chariot+nakazawa」はファイル名情報としてfilename1、プリンタアサイン情報としてclass1を使用する。この場合、filename1で定義したようにファイル名の1文字目をクラス情報として解釈し、class1で定義してあるようにBなら67901ps1へ送信するものと解釈する。例えば、最初のエントリの「chariot+nakazawa」から送られてきた「BTest」というファイルはプリンタ67901ps1に対して「Test」というファイル名で送信することを意味する。FTPクライアント、XNSサーバ等の他の機能においても同様な定義となる。

【0022】次に、本実施例の動作について説明する。アクセス可能なホストユーザは、前述のようにコネクションタイプ毎の情報ファイル、即ちスプールコネクション情報ファイル（ファイル名/etc/gw/hosts）411、FTPコネクションコネクション情報ファイル（ファイル名/etc/gw/ftphosts）412、XNSコネクションコネクション情報ファイル（ファイル名/etc/gw/xnshosts）413にあらかじめ定義され、また、送信側のプリンタコ

クションはプリンタコネクション情報ファイル414に定義され、それぞれコネクション情報記憶部41に記憶されている。まず、各コネクション情報ファイル411～414をファイル読み込み部421によってメモリ422に読み込む。読み込みに際しては、ファイルに有効なエントリが記述されていれば、そのコネクションタイプをサポートするものと認識して、記述されたエントリ順にコネクション識別子（コネクションID）を1から順に振ってメモリ上に展開していく。送信側のプリンタコネクション情報も記述されたコネクションIDを1から振って、メモリ上に展開する。図10は、FTPコネクション情報の場合のメモリ上の展開例を示すもので、エントリごとに順にコネクションIDが付与されている。

【0023】図11はゲートウェイのFTPサーバ機能、XNSサーバ機能へのホストからの接続要求に対する受付処理の概略を示すフロー図である。あるホストコンピュータからFTP接続要求があると、FTPコネクション受信処理部43は、その要求情報をアクセスチェック部424へ渡しアクセスチェックを依頼する（ステップS111）。そうするとアクセスチェック部424はメモリ422のFTPコネクション情報ファイルを参照して接続要求をしたホスト・ユーザがアクセス可能なホスト・ユーザに該当するか否かを調べる（ステップS112）。その結果、アクセス可能でなかったときはアクセスを拒否する（ステップS113）。アクセス可能と判定された場合には、アクセスチェック部424はコネクション接続要求のホスト名・ユーザ名を基に対応するコネクションIDをメモリ421から取り出し、FTPコネクション受信処理部43へ返す。FTPコネクション受信処理部43は、情報獲得部425を介して、コネクションIDを引数に、メモリ422から必要なコネクションに関する情報を得る（ステップS114）。そしてファイル受信処理を行う（ステップS115）。XNS接続要求があった場合も、同様に図11の手順でアクセスチェックをして、受付処理を行う。

【0024】FTPクライアント部による受信処理

FTPクライアント部14のFTPクライアント機能による受信処理に際しては、FTPコネクション情報の各コネクションについて受信プロセスを生成して、ホスト名およびユーザ名を得て接続を行う。図12はFTPクライアント機能の処理フローを示す図である。ゲートウェイのFTPクライアント14はファイル転送用のコネクションの確立要求をホストに対して行い、それが受理されるとファイル転送用のコネクションを確立する（ステップS121）。そして、ホストにログオンする（ステップS122）。これでホスト上のファイルを操作することが可能になる。次に、ディレクトリを変更するコマンドcdによりルートディレクトリからスプールディレクトリへ移動する（ステップS123）。そして、F

FTPクライアント部14はコマンドNLSTを発行してスプールディレクトリ中に受信要求のあるファイル(受信ファイル)があるかチェックする(ステップS124)。そのチェックの結果、受信ファイルがあればFTPクライアント部14はコマンドRETRを発行してそのファイルを受信し、それをスプール19に蓄積する(ステップS126)。前記チェックの結果、受信ファイルがなければ一定時間ウェイトし、ステップS124の先頭へもどる(ステップS125)。受信したファイルに関し、受信後すぐにホスト上のファイルを削除していくものかチェックする(ステップS127)。そのチェックの結果、Yesの場合、ホスト上のスプールディレクトリのファイルを削除する(ステップS128)。チェックの結果、Noの場合、ファイルをすべて受信したかどうかチェックする(ステップS129)。Noの場合、即ちまだ受信が終了していないファイルがあるときはステップS126の処理へすすみファイルの受信を続ける。Yesの場合、即ちすべてファイルを受信したときはステップS124の処理へ戻り待機する。

#### 【0025】FTPサーバ部による受信処理

図11により前述したように、FTPコネクションはホストからFTP接続要求があればFTPサーバ部15に生成されるプロセスである。FTP接続が確立された後は、FTPサーバ部15はホストから送られてくるFTPコマンドの処理を行う。FTPコネクションを用いてホストがプリントジョブをプリントシステムに送信する場合、FTPコマンドのSTORを用いる。コマンドの形式は

- 1) STOR /printer\_name/file\_name
- 2) STOR file\_name
- 3) STOR /class\_name/file\_name

の3通りがある。

- 1) は出力希望プリンタ名を指定する方法である。
- 2) はあらかじめ定義されているプリンタ指定文字をファイル名に組み込むことにより、出力プリンタを指定する方法である。
- 3) はあらかじめ定義されているクラス指定文字を指定することにより、出力希望プリンタを指定する方法である。

FTPコマンドにおける送信を表すコマンドであるSTORを用いた一連の処理の流れを図13、図14および図15に示す。図13に示すように、サーバの初期化処理(ステップS1301)、利用ホストは認可ファイルに登録されたものか調べるホストチェック(ステップS1302)を行い、チェックにパスしなかったときはエラー応答コードを返す(ステップS1303)。パスしたときにはFTP接続正常終了の応答コードを返し(ステップS1304)、コマンドの処理に移る。なお、これ

らのステップは図4により説明したコネクション管理により行うことができる。STORに先駆け他のFTPコマンド、即ち、FTPコマンドのUSERに関する処理を行うUSER処理(ステップS1305)、FTPコマンドのPASSに関する処理を行うPASS処理(ステップS1306)、FTPコマンドのSTRUに関する処理を行うSTRU処理(ステップS1307)、FTPコマンドのMODEに関する処理を行うMODE処理(ステップS1308)、FTPコマンドのTYPEに関する処理を行うTYPE処理(ステップS1309)、FTPコマンドのPORTに関する処理を行うPORT処理(ステップS1310)等の実行が必要であるが、それらのコマンドの内容は周知のものであるので説明は省略する。FTPコマンドのSTORに関する処理を行うSTOR処理(ステップS1311)については、図14および図15により詳細に説明する。

【0026】まず、FTPサーバ部15は、図14において、受信許可がないかをチェックし(ステップS1401)、受信許可がない場合には、エラー応答コードを返す(ステップS1402)。受信許可がある場合には、スプール領域が超過して使用できない状態になっているかチェックし(ステップS1403)、超過している場合にはエラー応答コードを返す(ステップS1404)。スプール領域に空きがあり使用可能な状態のときは、STORコマンドの引数部分をチェックし、それがプリンタを指定するものか、ファイル名を指定するものか、クラスを指定するものかを判定する(ステップS1405)。プリンタを指定するものであったときは、そのプリンタ名から内部ファイル名を獲得する(ステップS1406)。この場合、前述のように、

STOR /printer\_name/file\_name

の形式のコマンドであり、引数のprinter\_nameとfile\_nameから内部ファイル名を得ることができる。前記判定の結果、ファイル名を指定するものであるときは、そのファイル名から内部ファイル名を獲得する(ステップS1407)。この場合、前述のように、

STOR file\_name

の形式のコマンドであり、その引数であるfile\_nameの文字列の特定のあらかじめ定められた位置にプリンタを指定する文字が組み込まれており、その指定情報は前述のようにファイル名情報ファイルに記述されている。ファイル名情報ファイルからプリンタ指定文字の位置を得ることによりプリンタ名を決定し、file\_nameの残りの文字からファイル名を得ることにより、内部ファイル名を獲得することができる。例えば、file\_nameの文字列の5番目の文字がプリンタを指定する文字であるとファイル名情報ファイルに記述されているとき、file\_nameがAAAABAで

あれば、文字Bよりプリンタ名を決定し、残りの文字のAAAAAよりファイル名を決定し、これらから内部ファイル名を獲得することができる。また、前記判定の結果、STORコマンドの引数部分がクラスを指定するものであったときは、そのクラス名から内部ファイル名を獲得する(ステップS1408)。この場合、前述のように、

STOR /class\_name/file\_name

の形式のコマンドであり、図9に示すようなプリンタアサイン情報名ファイルを参照してプリンタ名を決定でき、従って内部ファイル名を獲得することができる。

【0027】次に、受信対象ファイルであるか否かをチェックする(S1409)。これは例えばプリンタアサイン情報名ファイルのクラスの定義にリトリブの対象としないものを定義しておき、これに該当するか否かでチェックをする。チェックの結果受信対象ファイルでなかったときは、エラー応答コードを返し(ステップS1410)、受信対象ファイルであったときはファイルをオープンする(ステップS1411)。ファイルをオープンできなかったときはエラー応答コードを返す(ステップS1412)。ファイルをオープンしたら、ホストからのプリントデータを受信する(ステップS1413)プリントデータの受信が正常終了したかチェックし(ステップS1414)、受信に失敗したときは、ファイルをクローズし受信ファイルを削除し(ステップS1415)、エラー応答コードを返す(ステップS1417)。プリントデータの受信が正常終了したらファイルをクローズする(ステップS1416)。次に、前記ステップS1405と同様にして、STORコマンドの形式によりプリンタ指定であるか、ファイル指定であるか、クラス指定であるかを判定し(ステップS1418)、プリンタ指定であれば受信したファイルをプリンタ名を用いてキューイングし(ステップS1419)、ファイル指定であればファイル名を用いてキューイングし(ステップS1420)、クラス指定であればクラス名を用いてキューイングする(ステップS1421)。キューイングが正常終了すれば、正常終了応答コードを返し(ステップS1424)、キューイングに失敗すればエラー応答コードを返す(ステップS1423)とともに、受信ファイルを削除する(ステップS1425)。

【0028】FTPサーバ部15は、FTPコネクションを用いてプリントジョブの詳細一覧を得る機能をも有する。この場合はFTPコマンドのLISTコマンドを用いる。コマンドの形式は

- 1) LIST
- 2) LIST /printer\_name
- 3) LIST /printer\_name/file\_name

の3通りがある。

1) はプリンタシステムに接続されている全てのプリンタの詳細情報一覧を得る方法である。

2) は指定プリンタにキューイングされているプリントジョブの詳細情報一覧を得る方法である。

3) は指定プリンタ、指定プリントジョブ名の詳細情報を得る方法である。

LISTを用いた一連の処理の流れを図16および図17に示す。図16に示すようにLIST処理に先駆け他のFTPコマンドの実行が必要であるが、それらのFTPコマンドの機能は周知であるので説明は省略する。図17において、まずLISTコマンドの引数があるか否かをチェックする(ステップS1701)。前述のコマンド形式から明らかなように引数がない場合はプリンタシステムに接続されている全てのプリンタの詳細情報一覧を得ることを指定しているので、全プリンタの状態一覧を作成する(ステップS1702)。そして登録されているクラス名一覧をマージする(ステップS1703)。引数がある場合には、第1の引数のプリンタ指定が正しいか否かをチェックし(ステップS1704)、否であるときはエラー応答コードを返す(ステップS1717)。そのプリンタ指定が正しいものであるときは指定プリンタにキューイングされているプリントジョブのリストを作成する(ステップS1705)。第2の引数があるか否かをチェックする(ステップS1706)。第2の引数があるときには、前述のように指定プリンタにキューイングされているプリントジョブのうち指定のプリントジョブの詳細を得るものであるので、ステップS1705で作成したリスト内を第2の引数で指定されているプリントジョブ名をキーとして検索する(ステップS1707)。検索した結果引数に指定したファイルがなかったときは、エラー応答コードを返す(ステップS1708、1717)。ステップS1703、ステップS1705、またはステップS1707で得られたリストのファイルをオープンする(ステップS1709)。オープンができたならファイル内容を送信する(S1713)。送信が正常終了したときは正常終了応答コードを返す(ステップS1716)。送信に失敗したらエラー応答コードを返す(ステップS1715)。

【0029】FTPサーバ部15のFTPサーバ機能によりプリントジョブ名を得ることもできる。この場合はFTPコマンドのNLSTコマンドを用いる。コマンドの形式は、

- 1) NLST
- 2) NLST /printer\_name
- 3) NLST /printer\_name/file\_name

の3通りがある。

1) はプリンタシステムに接続されている全てのプリン

タ名を得る方法である。

2) は指定プリンタにキューイングされている自ホスト、ユーザが送信し、プリンタのスプールキューにキューイングされているプリントジョブの名前一覧を得る方法である。

3) は指定プリンタ、指定プリントジョブ名の名前一覧を得る方法である（ワイルドカードを用いた検索が可能）。

NLS Tを用いた一連の処理の流れを図18および図19に示す。図18に示すようにLIST処理に先駆け他のFTPコマンドの実行が必要であるが、それらのFTPコマンドの機能は周知であるので説明は省略する。図19において、まずNLS Tコマンドの引数があるか否かをチェックする（ステップS1901）。前述のコマンド形式から明らかなようにNLS Tコマンドに引数がない場合はプリンタシステムに接続されている全てのプリンタ名を得ることを指定しているので、全プリンタ／クラス名の一覧を作成する（ステップS1902）。引数がある場合には、第1の引数のプリンタ指定が正しいか否かをチェックし（ステップS1903）、否であるときはエラー応答コードを返す（ステップS1915）。そのプリンタ指定が正しいものであるときは指定プリンタのスプールキューにキューイングされているプリントジョブ一覧を作成する（ステップS1904）。次に、第2の引数があるか否かをチェックする（ステップS1905）。第2の引数があるときには、前述のように指定プリンタにキューイングされている自ホスト、ユーザが送信し、プリンタのスプールキューにキューイングされているプリントジョブの名前一覧を得るものである。ステップS1904で作成したリスト内を第2の引数で指定されたプリントジョブ名で検索する（ステップS1906）。検索した結果引数に指定したファイルがなかったときは、エラー応答コードを返す（ステップS1907、1915）。ステップS1902、ステップS1904、またはステップS1906で得られたリストのファイルをオープンする（ステップS1908）。オープンができたならファイル内容を送信する（S1911）。送信が正常終了したときは正常終了応答コードを返す（ステップS1914）。送信に失敗したらエラー応答コードを返す（ステップS1913）。

#### 【0030】XNSサーバ部による受信処理

図20はXNXサーバ部16のXNXサーバ機能の処理の概略を示す図である。XNXサーバ部16はリモートホストからの要求があるかどうかチェックする（ステップS2001）。要求があれば、要求が何かを判定し各処理を行いリモートホストに応答を返し（ステップS2003～S2007）、なければ処理要求があるまでウェイトする（ステップS2002）。要求には、以下の4つがある。詳細は後述する。

#### ① プリント処理

リモートホストからのプリント要求を受け付ける（ステップS2004）。

#### ② プリント処理結果問合せ

プリント要求したものの処理結果を返す（ステップS2005）。

#### ③ プリンタ状態問合せ

ゲートウェイの状態（スプール可能か、処理中か等）を返す（ステップS2006）。

#### ④ プリンタ属性問合せ

ゲートウェイで受け付けられる属性情報（印字可能な用紙サイズ、両面印刷が可能か等）を返す（ステップS2007）。

#### 【0031】①プリント処理

図21はホストからのプリント処理要求に対するXNSサーバ部16の処理のフローを示す図である。XNSサーバ部16は、まず、プリント処理要求したホストが、登録されているホストであるか否かをチェックし（ステップS2101）、ホストが登録されていない場合はエラーを返す（ステップS2102）。次に、登録されているユーザかどうかチェックし（ステップS2103）、登録されていない場合はエラーを返す（ステップS2104）。次に、プロセスが動作状態かどうかチェックし（ステップS2105）動作状態でなければエラーを返す（ステップS2106）。次に、受信受付可能状態かどうかチェックし（ステップS2107）、受付可能状態状態でなければエラーを返す（ステップS2108）。さらに、スプール容量が足りているかどうかチェックし（ステップS2109）、スプール容量が不足していたらエラーを返す（ステップS2110）。また、キュー領域が足りているかどうかチェックし（ステップS2111）、キュー領域が不足していたらエラーを返す（ステップS2112）。次に、受信するファイル名の長さをチェックし（ステップS2113）、許容値を越える26文字以上ならエラーを返す（ステップS2114）。以上の各チェックを通過した場合には、ファイルを受信しはじめる（ステップS2115）。ファイル受信中に、書き込み失敗したらエラーを返す（ステップS2116、S2117）。受信に成功したら、ゲートウェイ10内の他の図示しない処理部にキューイングの依頼をする（ステップS2118）。キューイングの結果をチェックし（ステップS2119）、失敗したらエラーを返し（ステップS2120）、成功したらジョブ識別IDを返す（ステップS2121）。

#### 【0032】② プリント処理結果問合せ

図22はホストからのプリント処理結果問合せに対するXNSサーバ部16の処理フローを示す図である。XNSサーバ部16は、まず、要求のあったホストのホスト名が登録されているかどうかチェックし（ステップS2201）、ホスト名が登録されていない場合、エラーを返し、終了する（ステップS2202）。ホスト名が登

録されていない場合には、該当するジョブの処理結果をジョブ管理情報からチェックし（ステップS2203）、応答を返す（ステップS2204）。

#### 【0033】③ プリンタ状態問合せ

図23はホストからのプリンタ状態問合せに対するXNSサーバ部16の処理のフローを示す図である。ホスト名が登録されているかどうかチェックし（ステップS2301）、ホスト名が登録されていない場合、エラーを返し終了する（ステップS2302）。ホスト名が登録されていた場合には、ディスク使用率を調べ（ステップS2303）、スプール容量を超過していたらスプール状態=不可にセットする（ステップS2304）。そうでなければスプール状態=可にセットする（ステップS2305）。次に、接続されているプリンタがすべて停止状態か否かをチェックし（ステップS2306）、すべて停止状態の場合、フォーマッタ、プリンタ=不可にセットする（ステップS2307）。そうでなければフォーマッタ、プリンタ=可にセットする（ステップS2308）。受付可能な用紙サイズ=A4, A3, B4, B5をセットする（ステップS2309）。最後に、以上においてセットした事項を応答として返す（ステップS2310）。

#### 【0034】④ プリンタ属性問合せ

図24は、ホストからのプリンタ属性問合せに対するXNSサーバ部16の処理フローを示す図である。ホストが登録されているかどうかチェックする（ステップS2401）。ホスト名が登録されていない場合、エラーを返し終了する（ステップS2402）。ホスト名が登録されている場合、「両面印刷を受け付ける。ホチキス止めを受け付ける。受付可能な用紙サイズをA4, A3, B4, B5である。」というようにプリンタ属性情報を応答に設定する（ステップS2403）。そして、上記設定した応答を返す（ステップS2404）。

#### 【0035】XNS Printingクライアント部17による送信処理

図25は、XNS Printingクライアント部17の処理フロー図である。

1) 起動後、処理要求があるかどうかチェックする（ステップS2501）。

2-1) 要求がない場合、  
現在プリンタが停止中かビジー状態であるか使用可能かなどのプリンタの状態が現在どのようであるかを問合せ（ステップS2510）。その結果、プリンタが使用可能なら、既にプリント依頼をしてあるプリントジョブの処理が現在どのように進行しているか、即ち、プリント中であるか、プリントが終了しているか、待ち状態であるか、キャンセルされたかなどに関する問合せをする（ステップS2511、S2513）。

2-2) プリント要求がある場合、  
ステップS2501のチェックの結果プリント要求があ

る場合には、プリンタの状態を問合せ（ステップS2503）。問合せの結果、プリンタが使用可能なら結果問合せをする（ステップS2506）。送信ファイルがあればプリント要求を行う（ステップS2507、S2508）。

2-3) 停止要求がある場合、

ステップS2501のチェックの結果、停止要求がある場合には、停止処理を行いプロセスを終了する（ステップS2502）。

10 【0036】以上に詳述した第1の実施例は、受信部としてFTPクライアント機能とFTPサーバ機能のほか  
に更に、プリンタのサポートするプロトコルXNSを受け付けることのできるXNSサーバ機能を設けたので、  
プリンタに対し直接にプリント要求をしていたホストからもゲートウェイを利用することができるようになり、  
ホストの接続性がより向上した。また、XNSプロトコルによるプリント要求を直接にプリンタに与えるのではなくゲートウェイを介することにより、プリンタのス  
プール領域の制限、プリンタに同時に接続できるコネクションの制約、プリンタダウン時の受付不能等のプリンタ  
側の状態にプリント要求受付が左右されることがなくな  
った。また、本実施例は、コネクションの種類が増加  
しているため、従来の延長でコネクションの管理をおこな  
ったのでは、管理が複雑化し、処理が効率的でなくな  
ってしまうが、本実施例ではコネクションの管理をコネ  
クション識別子とコネクションタイプで管理するように  
したので、プロトコルによるコネクションの管理を効率的  
に行うことができる。また、そのコネクション識別子  
は、コネクションの管理のみならずロギング情報の書き  
込み・参照、履歴・アカウントの書き込み・参照、プ  
リント要求の参照・取り消しなどにも用いることができ  
る。例えば、コネクションIDに対応したファイルにロ  
ギング情報を書き込むことにより、コネクション毎に簡  
単に検索を行うことができる。なお、その際、定義して  
いないホストやユーザからのプリント要求は特殊コネ  
クションIDにもとめて格納すればよい。

40 【0037】（第2の実施例）従来の技術においてゲートウェイとプリンタ間で要求/データの交換をする場合、まずOS I 7階層の下位レイヤ（ネットワーク層）でコネクション（通信路）を開設し、そのコネクションを利用して上位レイヤのデータを交換し、データの交換が終了したら必ずコネクションを解放していた。この方式だと、ひんばんにデータの交換が行われる場合、そのたびに必ずコネクションの開設/解放を行わなければならない効率のよいデータ交換ができなかった。この実施例はこの問題を解決するものである。図26に示すようにこの実施例は、プロトコルFTPまたはプロトコルXNSのように異なるプロトコルを有するホストコンピュータ（単に「ホスト」という）が複数台261、  
～261n、プロトコルXNSを有するプリンタが複数

台262<sub>1</sub>~262<sub>2</sub>、およびプロトコルFTP、プロトコルXNS両方を有するゲートウェイ260が同一の論理ネットワーク263上に存在して構成されている。ゲートウェイ260は、第1の実施例と同じく、図4で説明したコネクション管理を行う管理部267を有するほか、タイマ268を用いた管理を行う。ホストとゲートウェイ間のファイル転送はホストのサポートする通信プロトコルFTP、TCP/IPを通して行い、FTPサーバ部264で受信処理し、ゲートウェイとプリンタ間のファイル転送は、プリンタのサポートする通信プロトコルXNS Printingで転送処理するXNS Printingクライアント部265およびその下位レイヤによって行う。ゲートウェイ260にはホストから受信したファイルを一時的に蓄積するスプール266がある。

【0038】図27にゲートウェイ、プリンタ間の処理手順のフローを示す。XNS Printingクライアント部265は、プリントシステムへ送信すべき処理があるかをチェックし(ステップS271)、あるときにはプリンタとのコネクションが確立されているか否かをチェックし(ステップS272)、確立されていなければ、その送信処理を実行し(ステップS273)、確立されていなければコネクション確立処理をした後(ステップS274)、その送信処理を実行する(ステップS275)。処理の実行が終了したらタイマ268の設定を行い(ステップS276)、タイマを始動させる。タイムアウトになったかをチェックし(ステップS277)、タイムアウトになっていなければ、コネクションの解放は行わず、タイムアウトになっていればコネクション解放処理をする(ステップS278)。なお、タイマの設定後、一定時間処理要求がなかった場合タイムアウトとなり、タイマの設定後、一定時間内に処理要求があった場合タイムアウトにはならない。

【0039】図28は本実施例の処理シーケンスの例を示し、図29は比較のために従来例の処理シーケンスの例を示す。ゲートウェイとプリンタの間では、コネクションを開設し、データ交換等の処理を行った後、コネクションの解放を行う。図29に示す従来のシーケンスでは、コネクションの開設を行いデータ交換等のひとつの処理を行うと必ずコネクションの解放を行っていた。即ち、ひとつの処理ごとにコネクションの開設と解放を必ず行っていた。そのため頻繁にデータの交換が行われる場合、そのたびに必ずコネクションの開設と解放を行うため効率のよいデータ交換ができなかった。本実施例では、図28に示すように、データの交換後にタイマを始動し、タイマに設定した一定時間を経過するとコネクションを解放し、その一定時間以内に次の処理要求があったときはタイマをリセットしその要求された処理を行う。従って、頻繁にデータ交換が行われる場合には一度開設したコネクションを何度も利用することができ、効

率よくデータの交換を行うことができる。

【0040】

【発明の効果】本発明(請求項1)によれば、コネクションの管理をコネクション識別子とコネクションタイプで管理するようにしたので、プロトコルによるコネクションの管理を効率的に行うことができる。

【0041】また、本発明(請求項2)によれば、コネクションが確立された後、所定時間内に次の処理要求があるときにはコネクションの解放を遅延させるようにしたので、頻繁にデータ交換が行われる場合には一度開設したコネクションを何度も利用することができ、効率よくデータの交換を行うことができる。

【0042】また、本発明(請求項3)によれば、プリンタのサポートする通信プロトコルによるプリント要求を受け付ける手段(XNSサーバ機能)を設けたので、プリンタに対し直接にプリント要求をしていたホストからもゲートウェイを利用することができるようになり、接続性が向上した。また、このようにプリント要求をゲートウェイを介して受け付けるようにしたことにより、プリンタのスプール領域の制限、プリンタに同時に接続できるコネクションの制約、プリンタダウン時の受付不能等のプリンタ側の状態にプリント要求受け付けが制限されることがなくなった。

【図面の簡単な説明】

【図1】 本発明の第1の実施例の構成の概略を示す図

【図2】 ホストからプリンタへのデータの流れを説明するための図

【図3】 ホストからプリンタへのデータの流れを説明するための図

【図4】 ゲートウェイのコネクション管理に関する構成を示す図

【図5】 スプールコネクション情報ファイル(FTPクライアント機能における接続ホスト記述ファイル)の記述形式および記述例を示す図

【図6】 FTPコネクション情報ファイル(FTPサーバ機能における接続ホスト記述ファイル)の記述形式および記述例を示す図

【図7】 XNSコネクション情報ファイル(XNSサーバ機能における接続ホスト記述ファイル)の記述形式および記述例を示す図

【図8】 プリンタコネクション情報ファイル(XNS Printingサーバ機能における接続ホスト記述ファイル)の記述形式および記述例を示す図

【図9】 プリンタアサイン情報名ファイルの記述例を示す図

【図10】 FTPコネクション情報のメモリへの格納形式を示す図

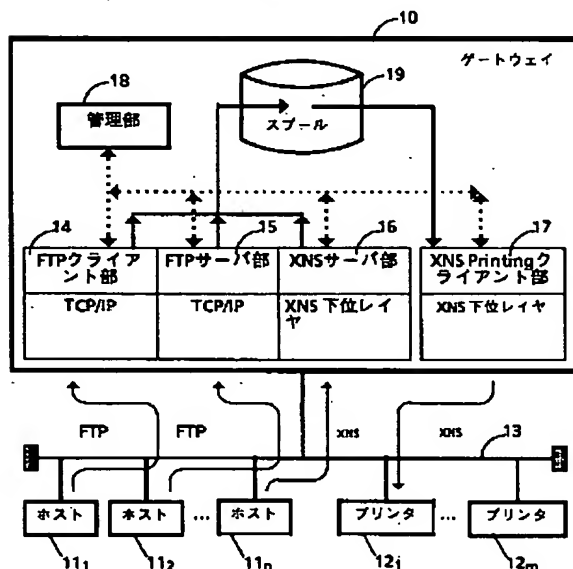
【図11】 ゲートウェイのFTPサーバ機能、XNSサーバ機能へのホストからの接続要求に対する受付処理の概略を示すフロー図



- 【図12】 FTPクライアント機能の処理フロー図  
 【図13】 FTPコマンドにおける送信を表すSTORを用いた一連の処理のフローを示す図(1)  
 【図14】 FTPコマンドにおける送信を表すコマンドであるSTORを用いた一連の処理のフローを示す図(2)  
 【図15】 FTPコマンドにおける送信を表すコマンドであるSTORを用いた一連の処理のフローを示す図(3)  
 【図16】 FTPコマンドのLISTを用いた一連の 10 処理のフローを示す図(1)  
 【図17】 FTPコマンドのLISTを用いた一連の処理のフローを示す図(2)  
 【図18】 FTPコマンドのNLSTを用いた一連の処理のフローを示す図(1)  
 【図19】 FTPコマンドのNLSTを用いた一連の処理のフローを示す図(2)  
 【図20】 XNSサーバ機能の処理フローを示す図  
 【図21】 プリント処理要求に対する処理のフローを示す図  
 【図22】 プリント処理結果問合せに対する処理のフ

【図1】

図1 第1の実施例



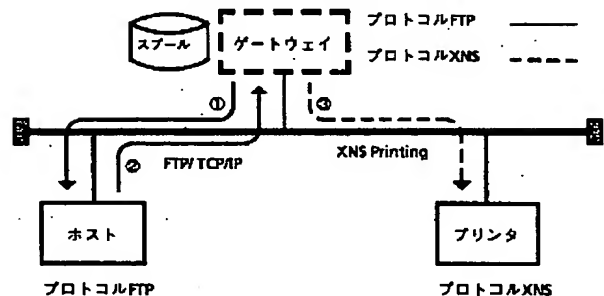
ローを示す図

- 【図23】 プリント状態問合せに対する処理のフローを示す図  
 【図24】 プリント属性問合せに対する処理のフローを示す図  
 【図25】 XNS Printingクライアント機能の処理のフローを示す図  
 【図26】 本発明の第2の実施例の構成の概略を示す図  
 【図27】 コネクションの開設・解放の処理フローを示す図  
 【図28】 第2の実施例の処理シーケンスの例を示す図  
 【図29】 従来の処理シーケンスの例を示す図  
 【符号の説明】  
 11<sub>1</sub>~11<sub>n</sub>…ホスト、12<sub>1</sub>~12<sub>n</sub>…プリンタ、13…ネットワーク、14…FTPクライアント部、15…FTPサーバ部、16…XNSサーバ部、17…XNS Printingクライアント部、18…管理部、19…スプール。

20

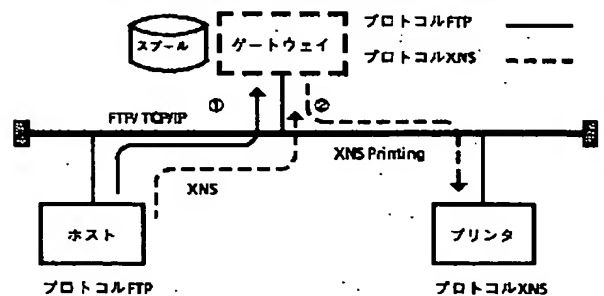
【図2】

図2 ホストからプリンタへのデータの流れ(#1)



【図3】

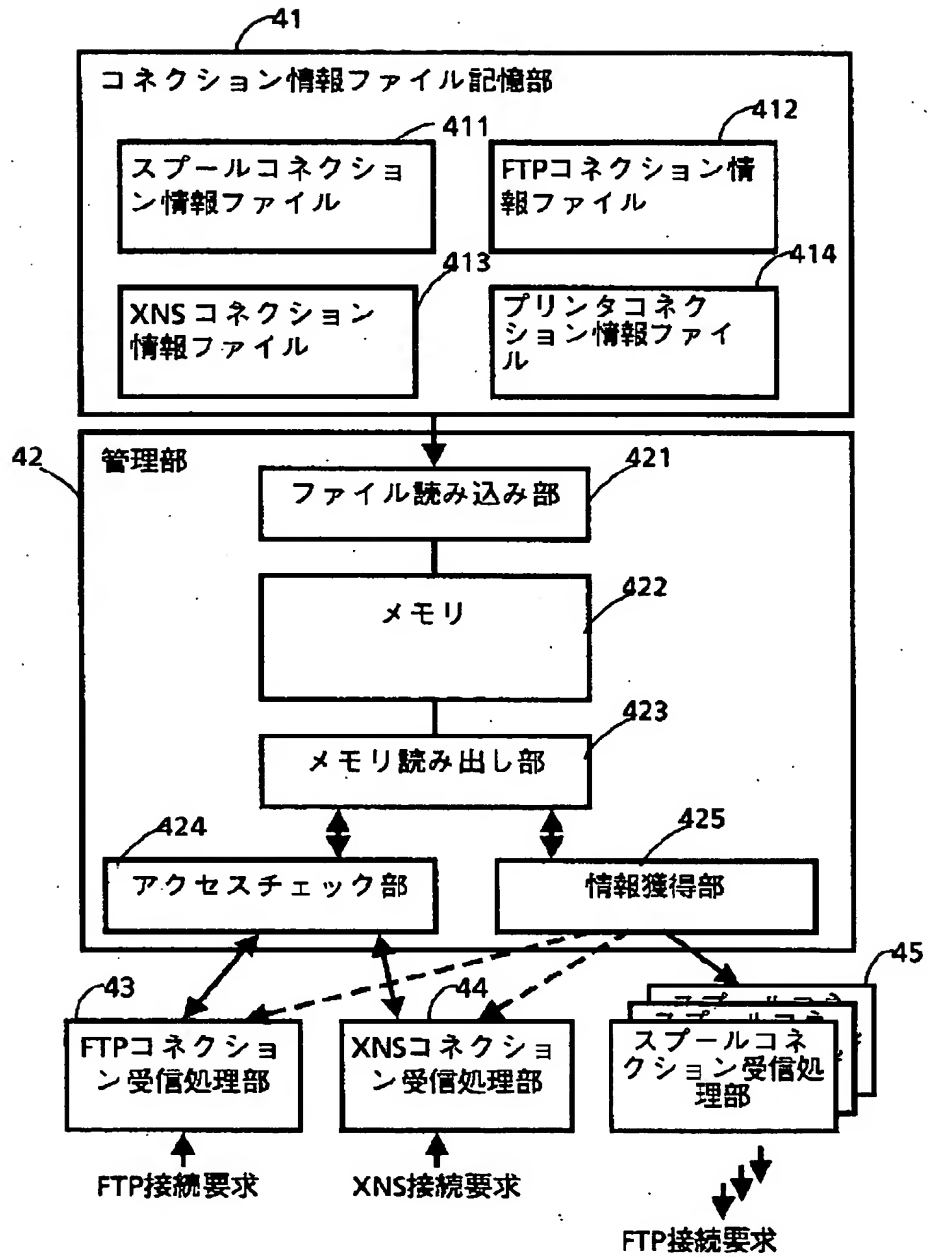
図3 ホストからプリンタへのデータの流れ(#2, #3)





【図4】

図4



【図5】

図5 スプールコネクション情報ファイル

## 【記述形式】

machine:type:login:password:account:spool:filename:printer\_\_assign

machine	ホスト名
type	ホストタイプ
login	ログインユーザ名
password	パスワード
account	アカウント
spool	スプール名
filename	ファイル名情報名
printer__assign	プリンタアサイン情報名

## 【記述例】

```
# /etc/gw/host
acos77:ACOS6:nakazawa:nakazawa:toshi:FXSPOOL:filename1:printer__assign2
acos66:ACOS4:kogane::kogane:kon:FXSPOOL10:filename1:printer__assign1
```

【図6】

図6 FTPコネクション情報ファイル

## 【記述形式】

hostname:username:filename:printer\_\_assign

hostname	ホスト名
username	ユーザ名
filename	ファイル名情報の名前
printer__assign	プリンタアサイン情報の名前

## 【記述例】

```
# /etc/gw/ftphosts
chariot:nakazawa:filename1:printer__assign1
supreme:ichikawa:filename2:printer__assign2
pansy:nakatani:filename3:printer__assign3
```

【図7】

図7 XNSコネクション情報ファイル

## 【記述形式】

address:hostname:username:filename:printer\_\_assign

address	アドレス
hostname	ホスト名
username	ユーザ名
filename	ファイル名情報の名前
printer__assign	プリンタアサイン情報の名前

## 【記述例】

```
# /etc/gw/xnshosts
00001cf7:080020001cf7:VAX780@1:vaxtaro:filename1:printer__assign1
:080037002939:xnshost7:username:filename2:printer__assign2
```

【図8】

図8 プリンタコネクション情報ファイル

## 【記述形式】

printername:type:address

printername	プリンタ名
type	プリンタタイプを記述
address	MACアドレス

## 【記述例】

```
# /etc/gw/printer
6790lps1:6790LPS:00001cf7:080037002939
6790lps2:6790LPS:00001cf7:08003700294a
6790lps7:6790LPS:00000000:080037003000
6790lps9:6790LPS::080037003000   アドレス記述省略(= 00000000)
```

【図9】

図9 プリンタアサイン情報名ファイル

<b>class1</b>					
B	6790 ps1				[class1の定義] プリンタアサイン情報の名前
C	6790 ps1, 6790 ps2				6790 ps1へ送信
F	ALL				6790 ps1, 6790 ps2へ送信
default	6790 ps9				接続プリンタすべてに送信
					デフォルトのプリンタとして6790 ps9を設定
<b>class2</b>					
A	6790 ps1				[class2の定義] プリンタアサイン情報の名前
default	6790 ps7				6790 ps1へ送信
					デフォルトのプリンタとして6790 ps7を設定

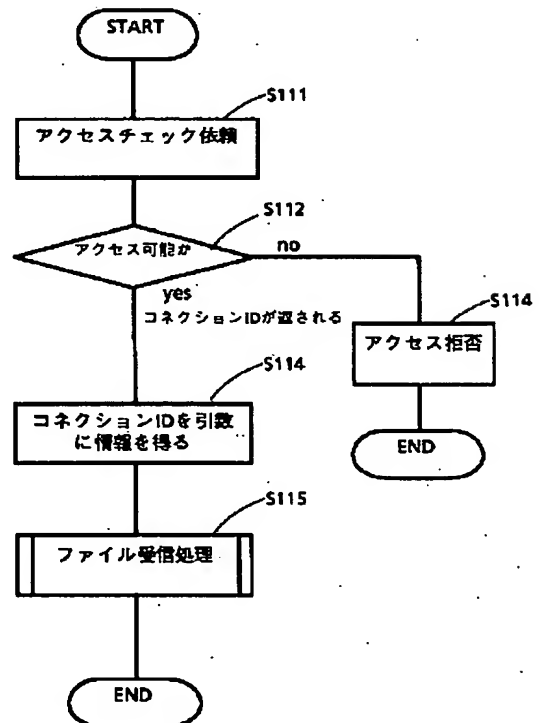
【図10】

図10 FTPコネクション情報

1	…コネクションID
chariot	…ホスト名
nakazawa	…ユーザ名
filename1	…ファイル名情報の名前
printer_assign1	…プリンタアサイン情報の名前
2	
supreme	
Ichikawa	
filename2	
printer_assign2	
3	

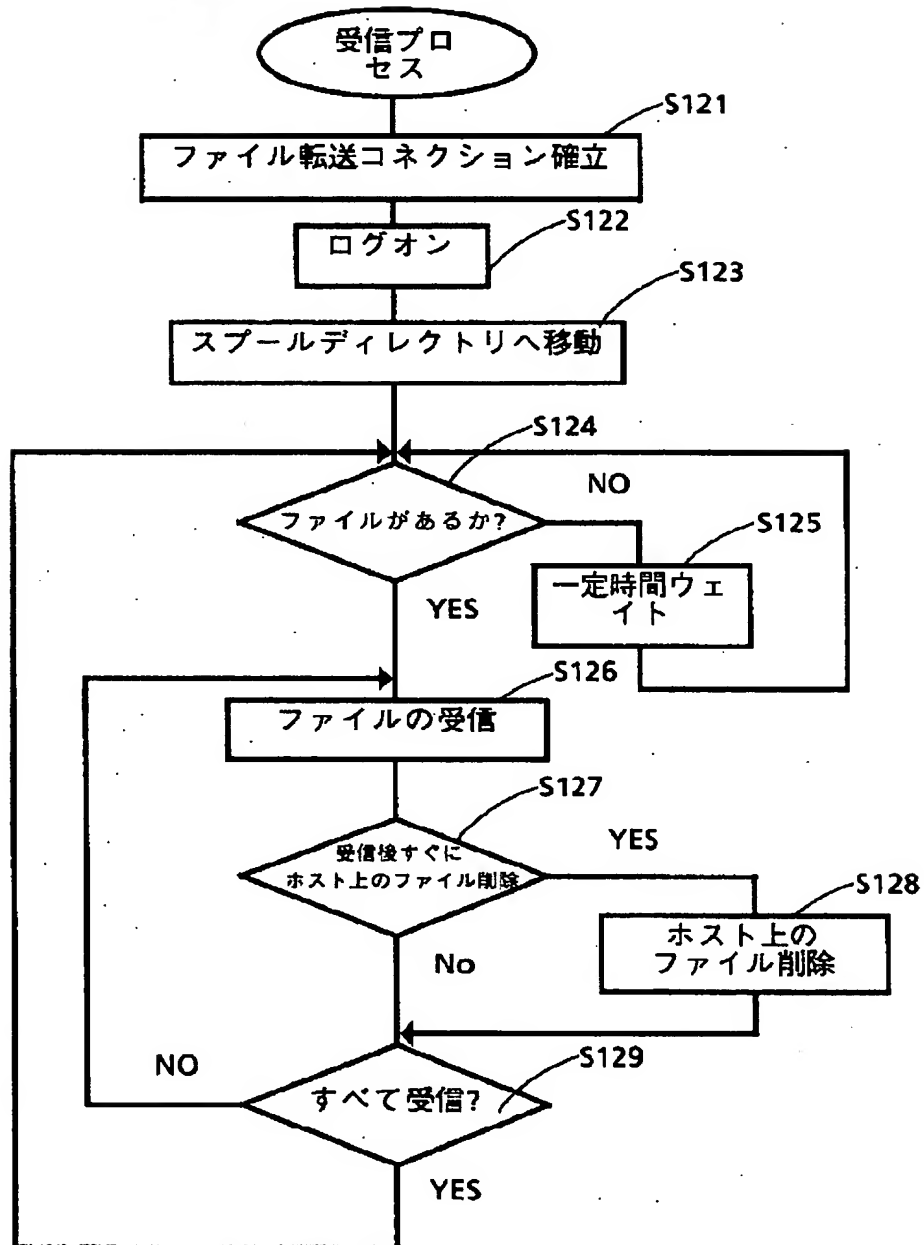
【図11】

図11 コネクション処理



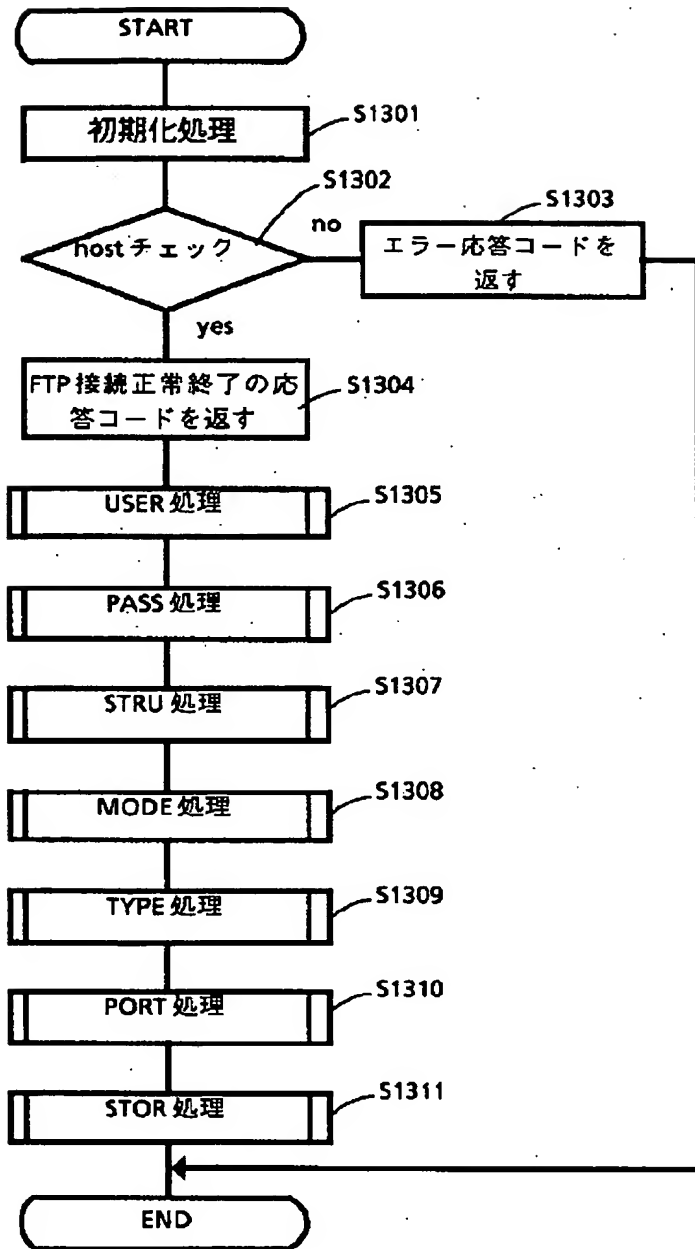
【図12】

図12 FTPクライアント機能の処理



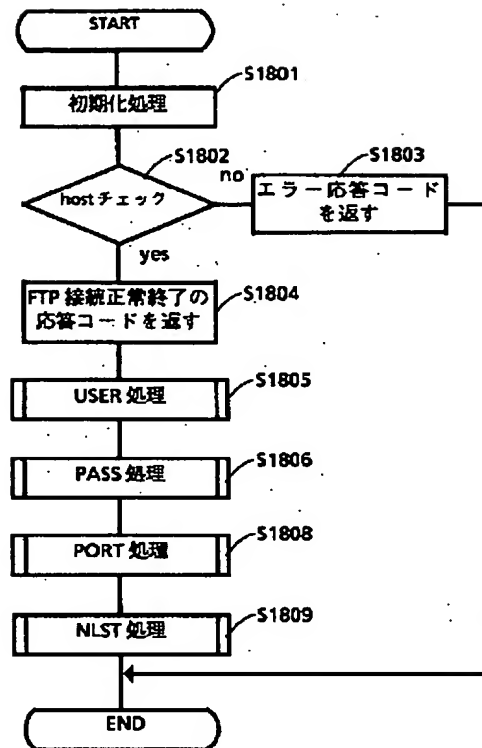
【図13】

図13 ファイル受信



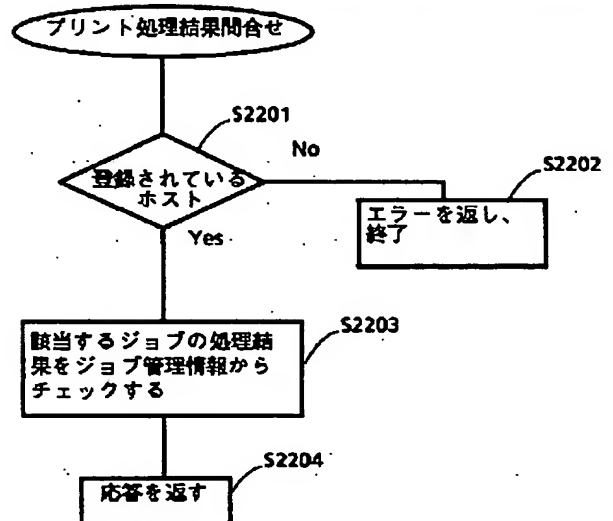
【図18】

図18 NLSTの処理(1)



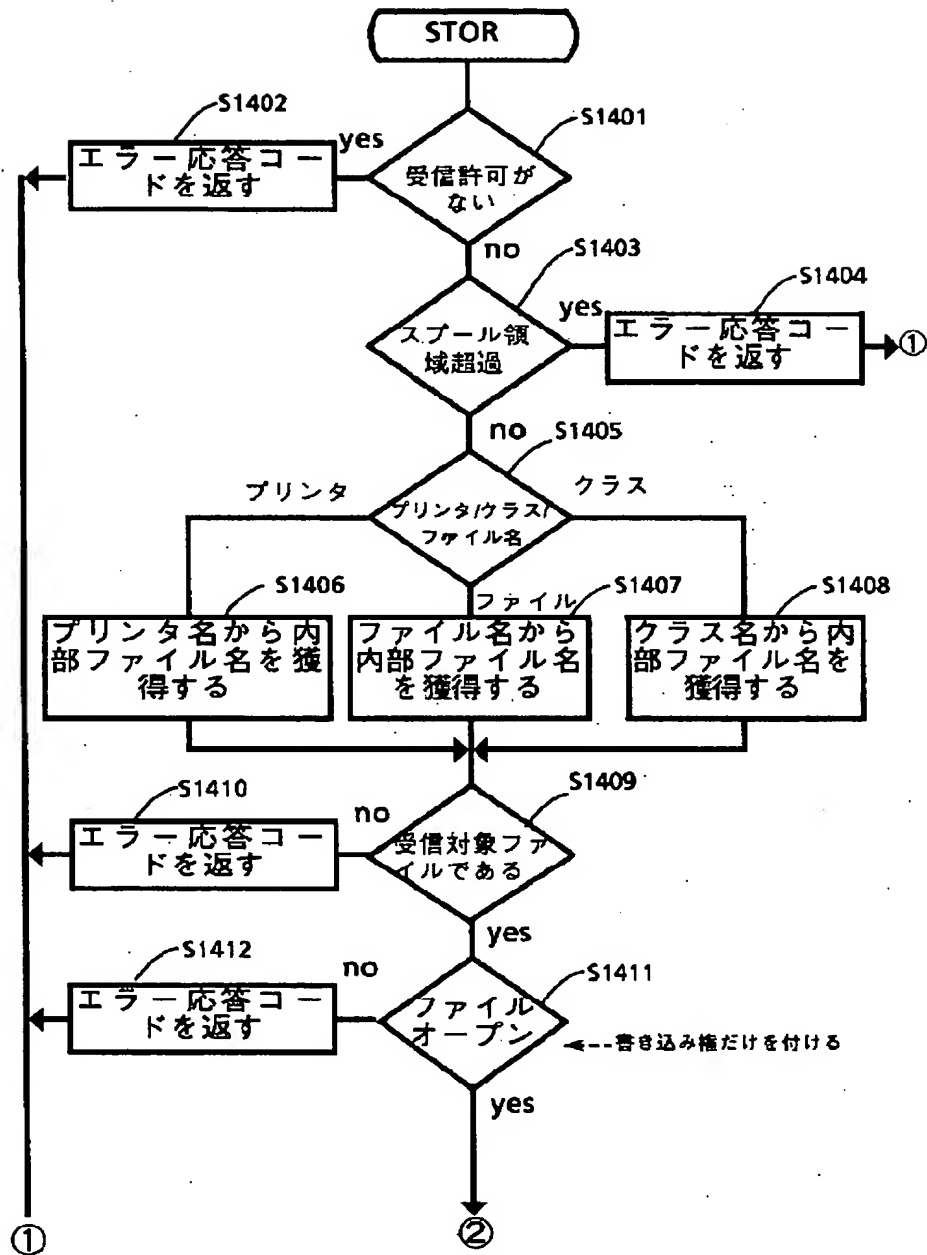
【図22】

図22 プリント処理結果問合せに対する処理



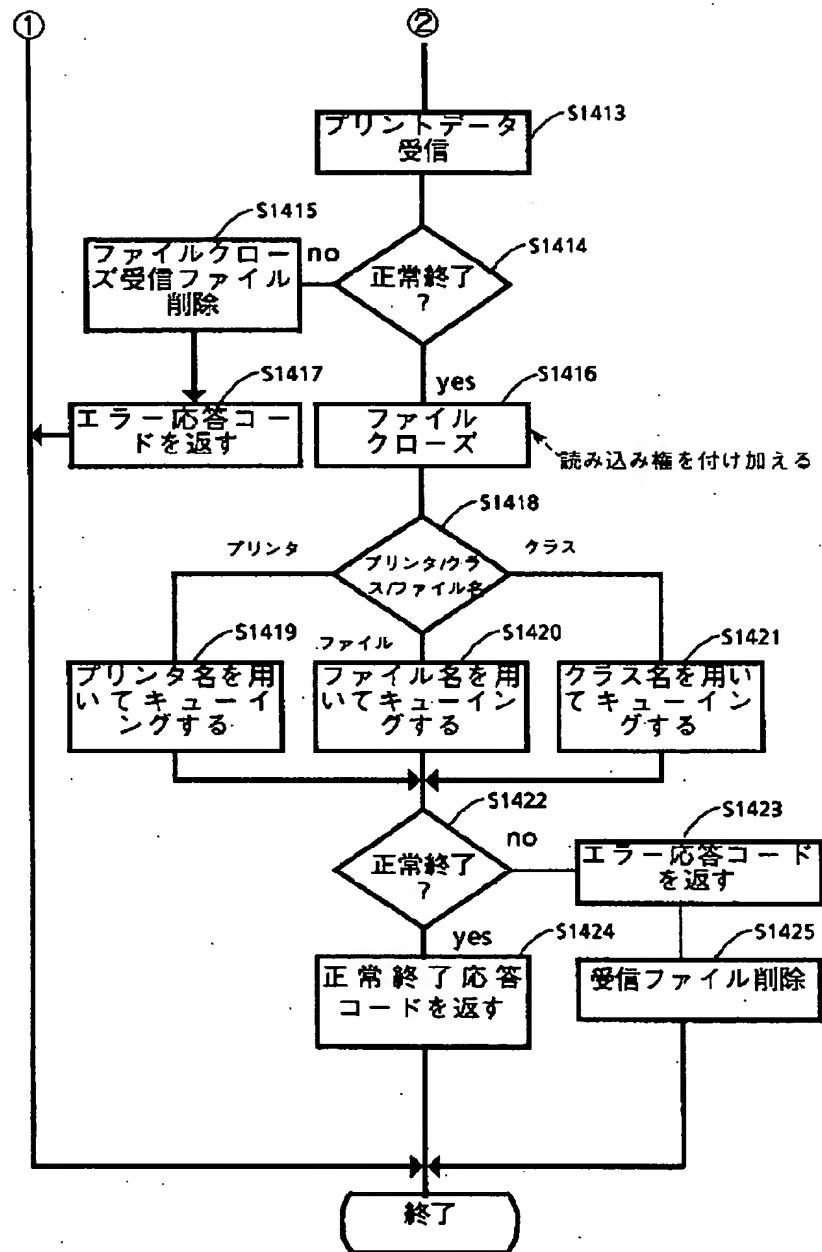
【図14】

図14 STORの処理(1)



【図15】

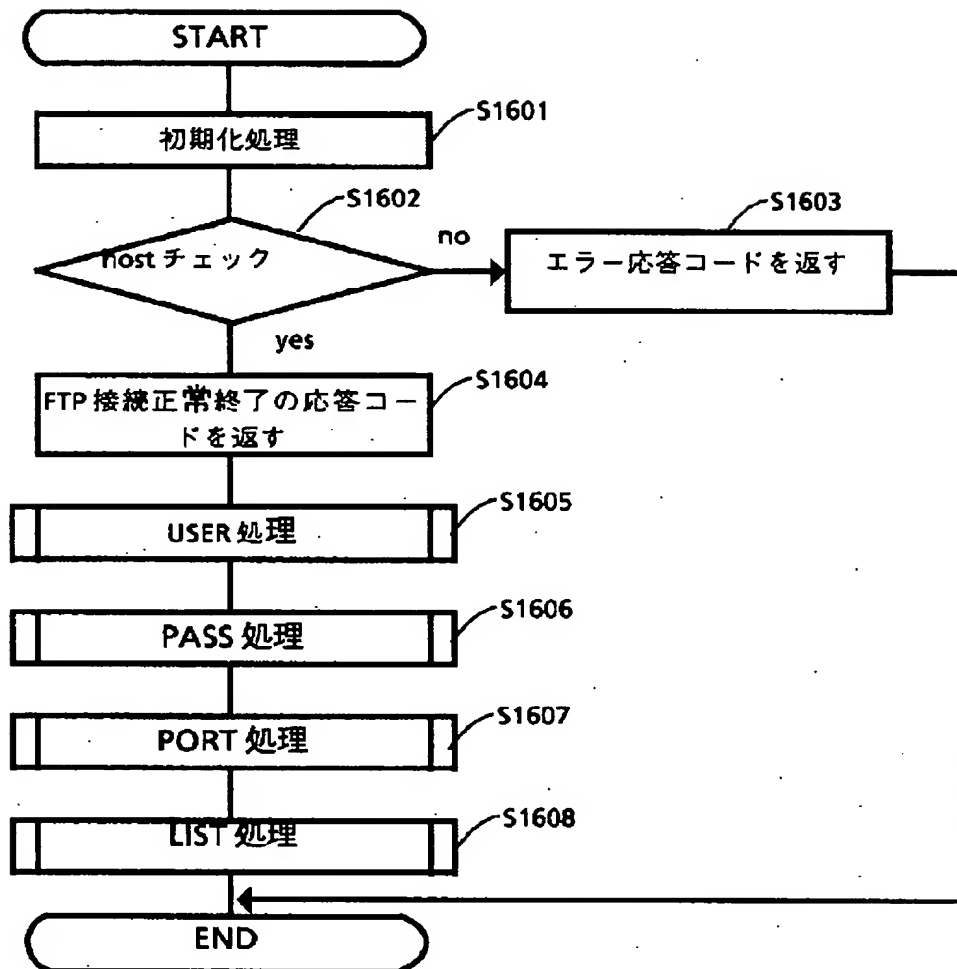
図15 STORの処理(2)





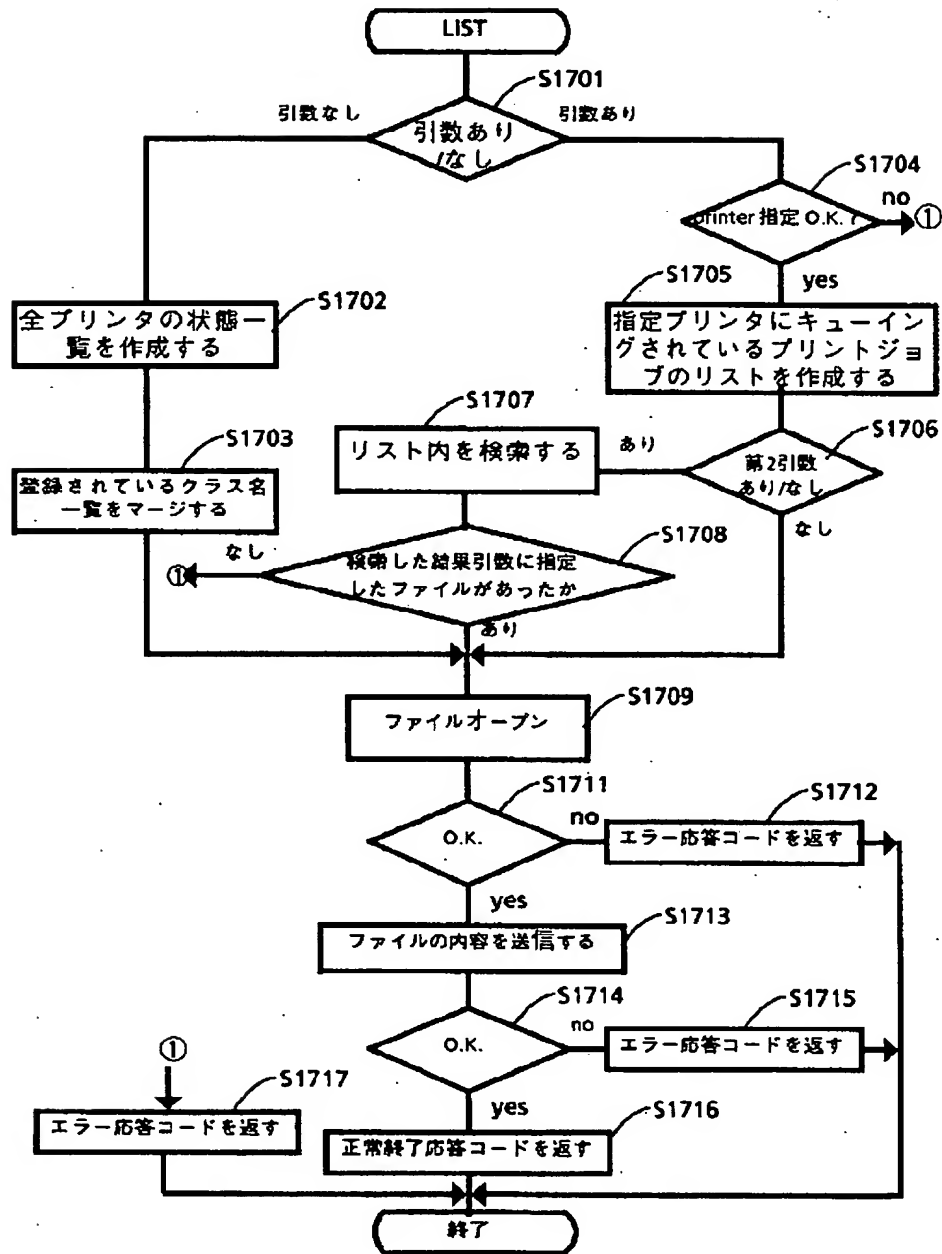
【図16】

図16 LISTの処理(1)



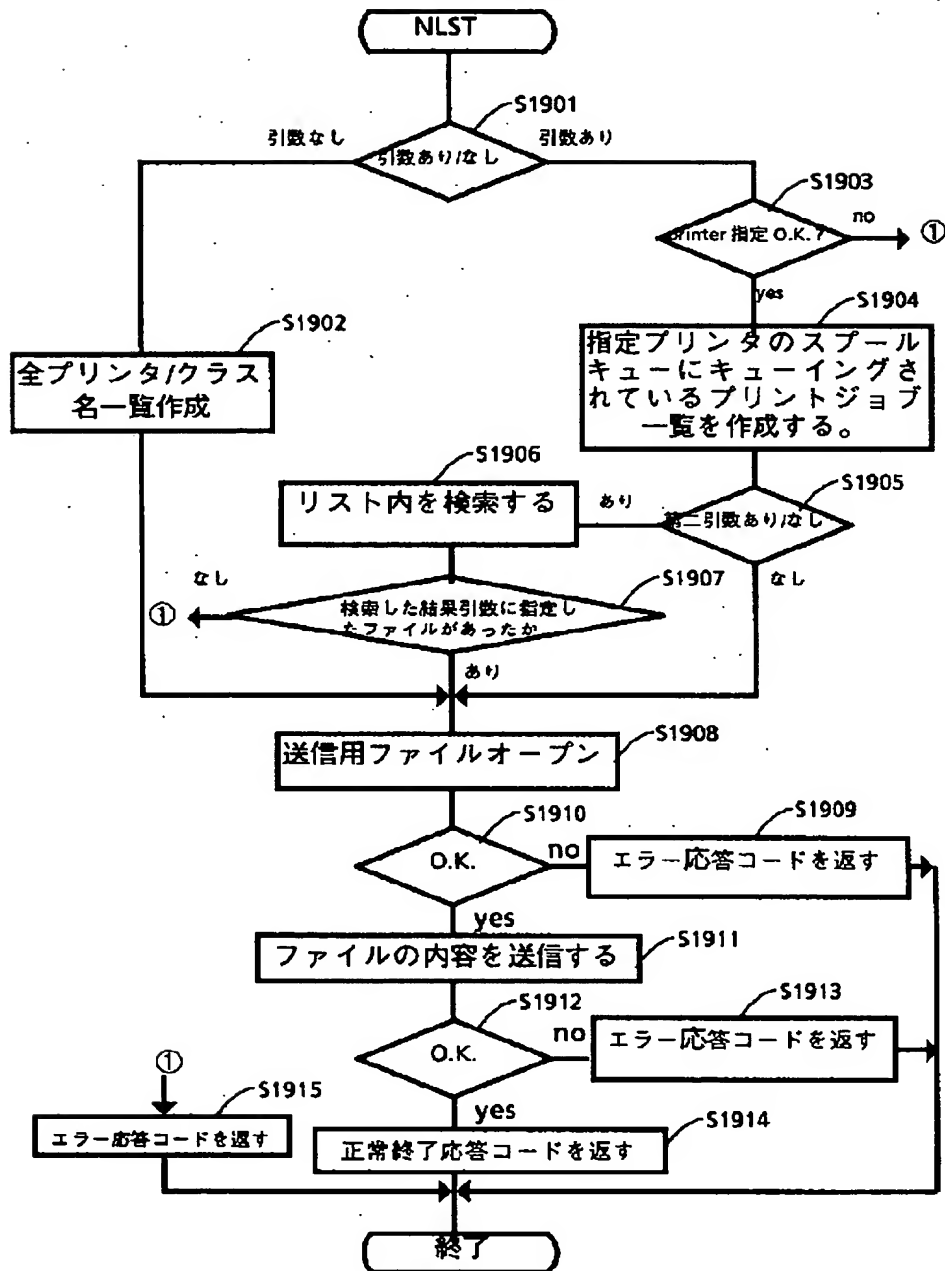
【図17】

図17 LISTの処理(2)



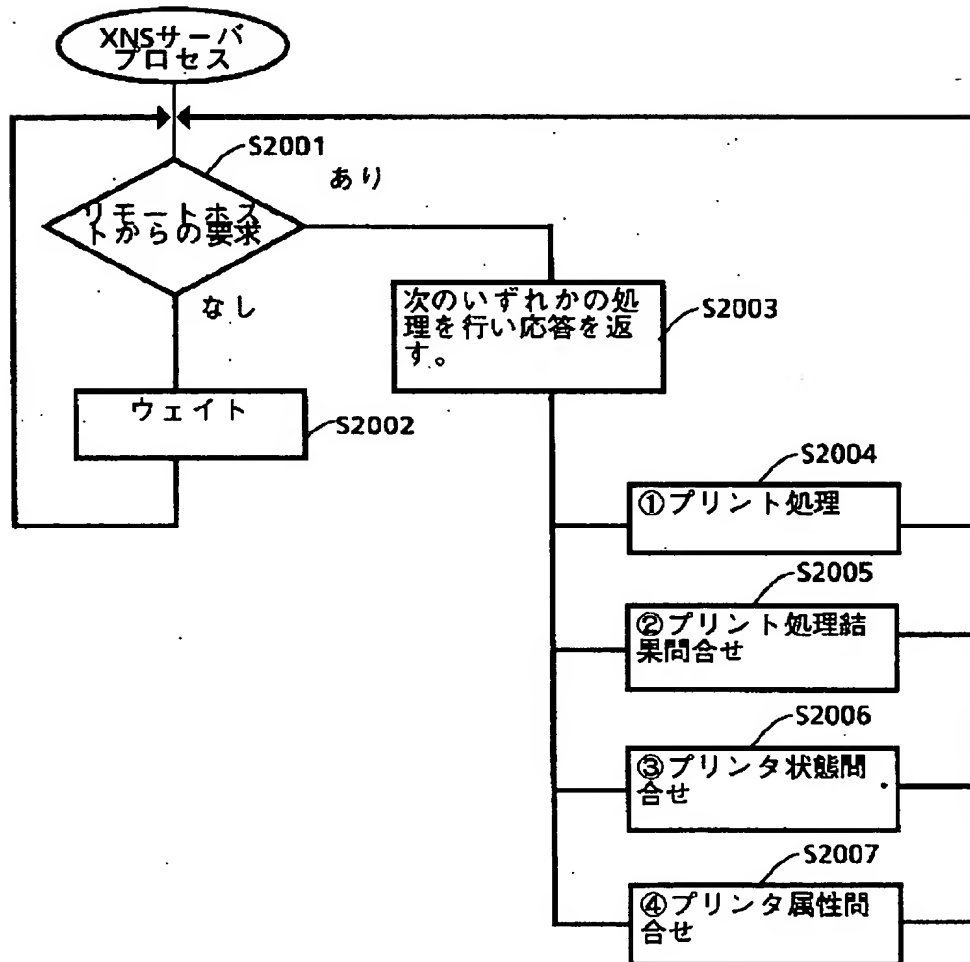
【図19】

図19 NLSTの処理(2)



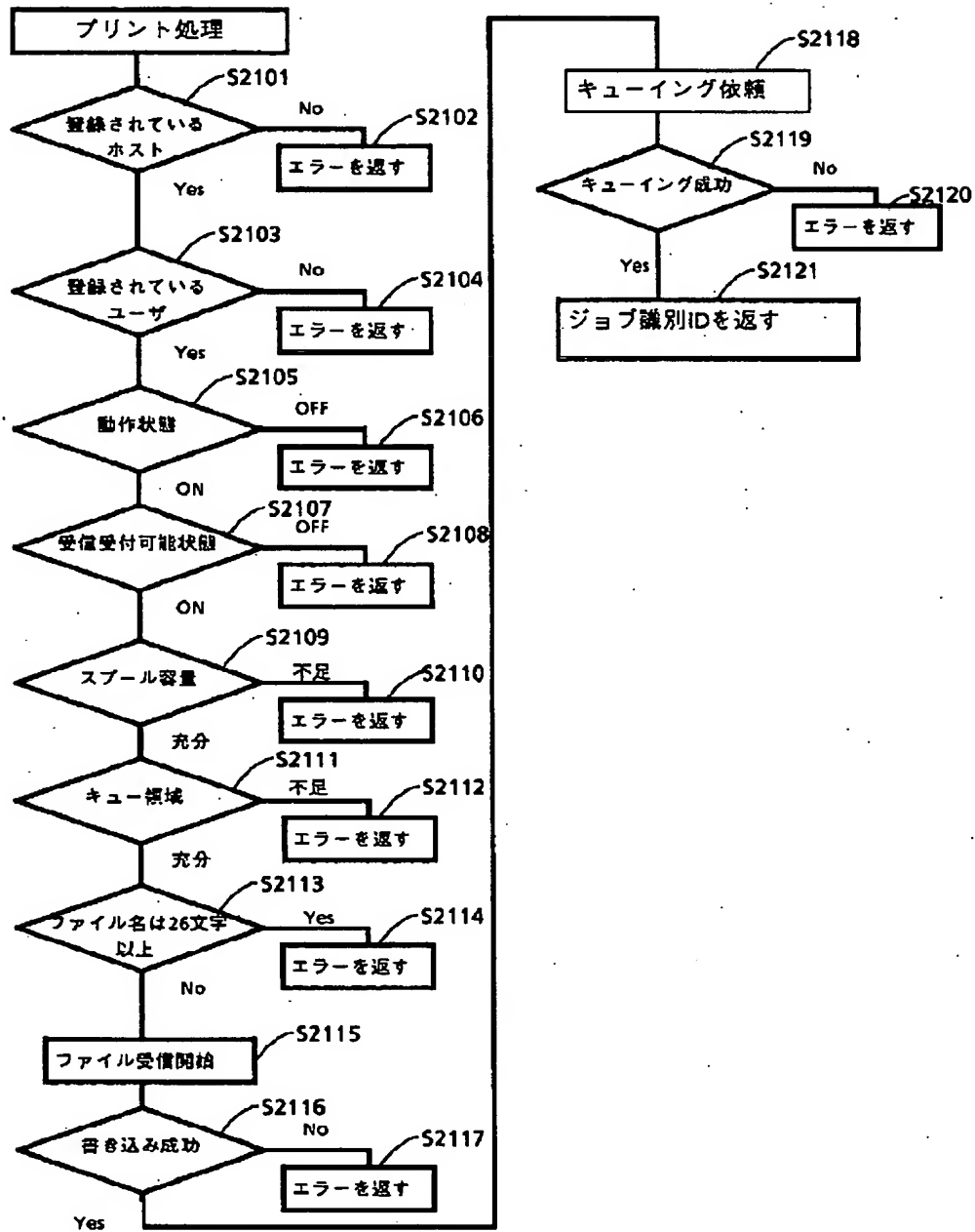
【図20】

図20 XNSサーバ機能の処理



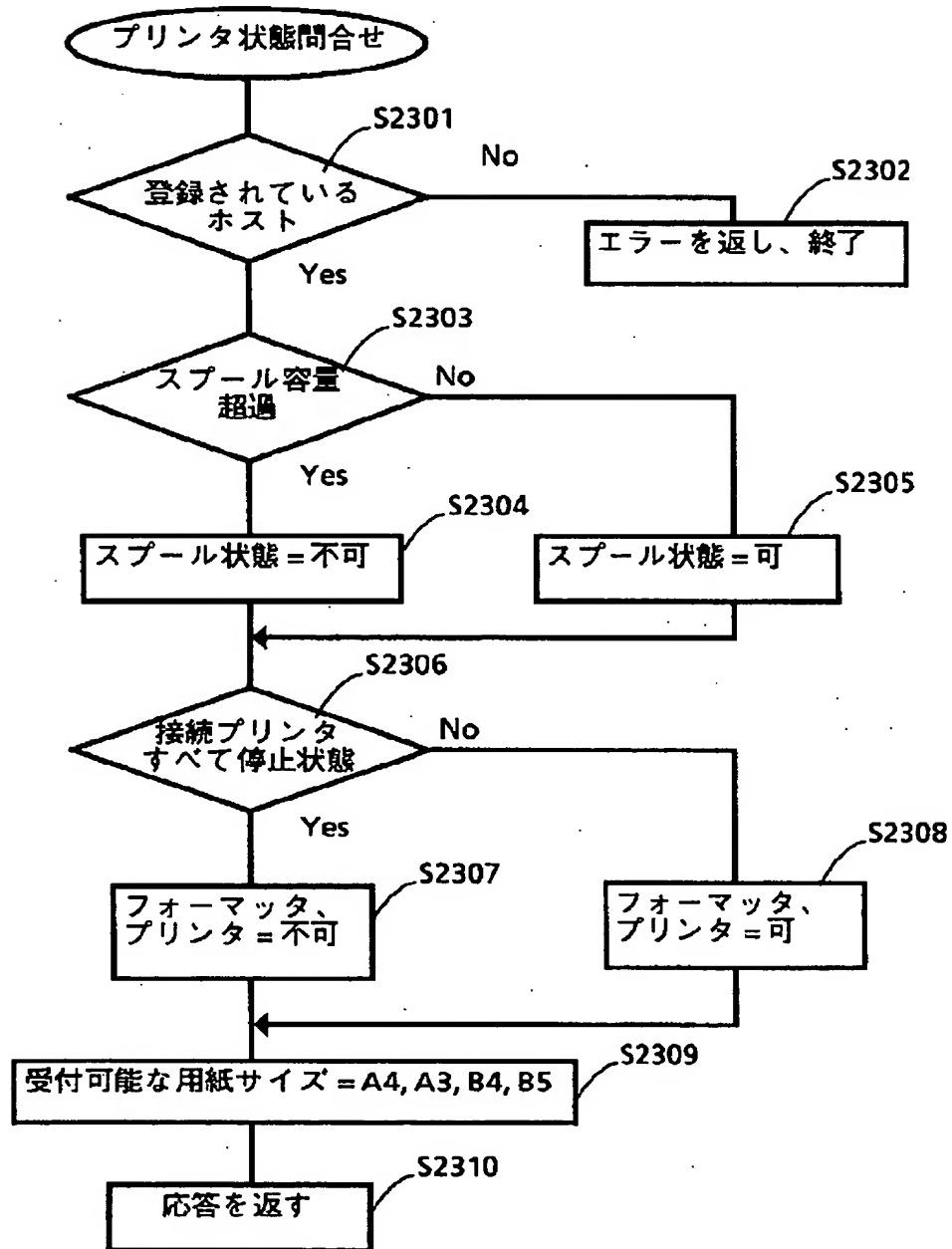
【図21】

図21 プリント処理要求に対する処理



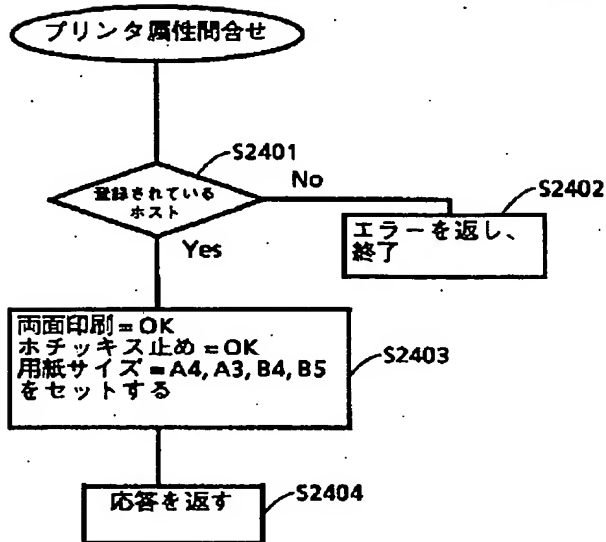
【図23】

図23 プリンタ状態問合せに対する処理



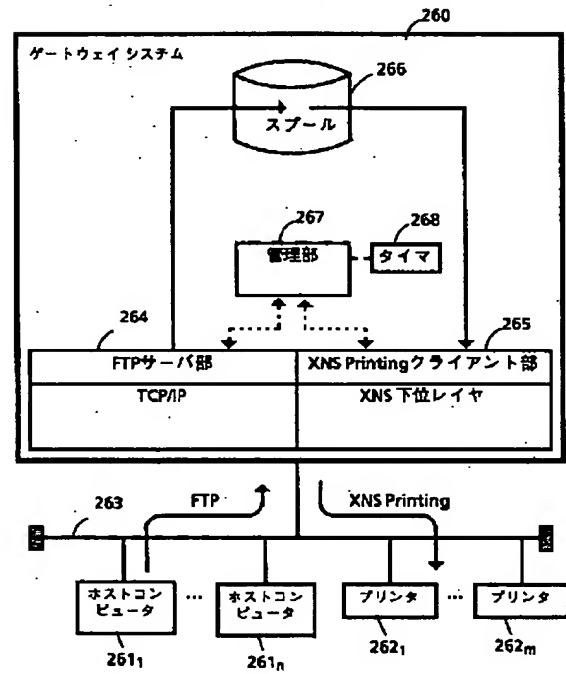
【図24】

図24 プリンタ属性問合せに対する処理



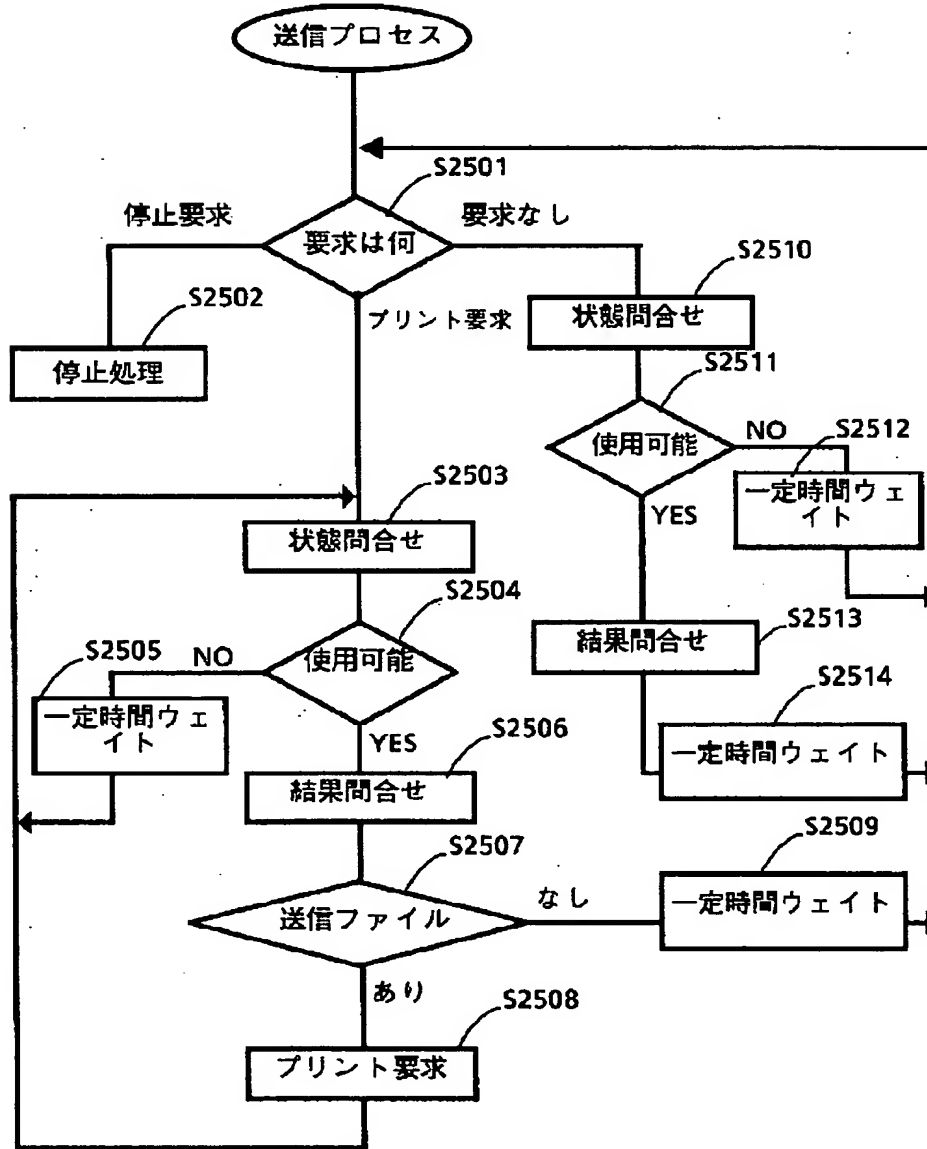
【図26】

図26 第2の実施例



【図25】

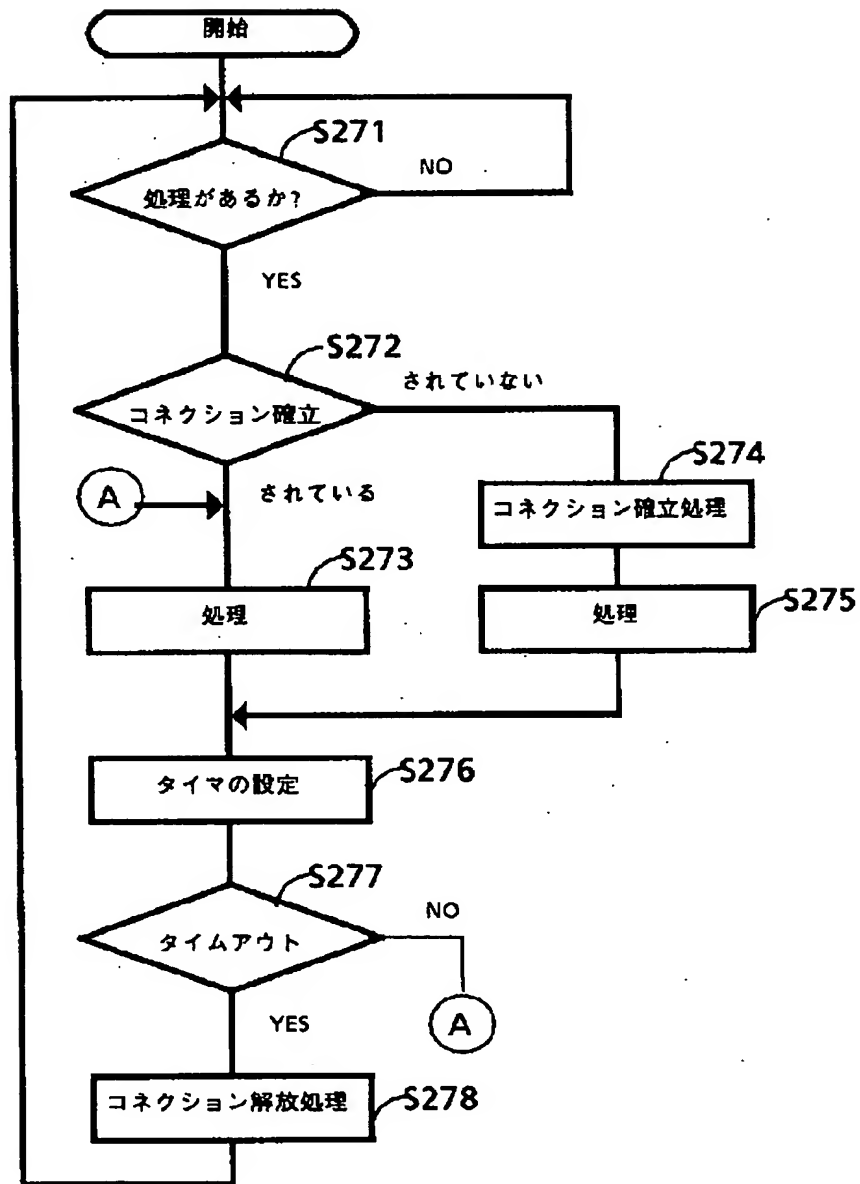
図25 XNS Printingクライアント機能の処理フロー





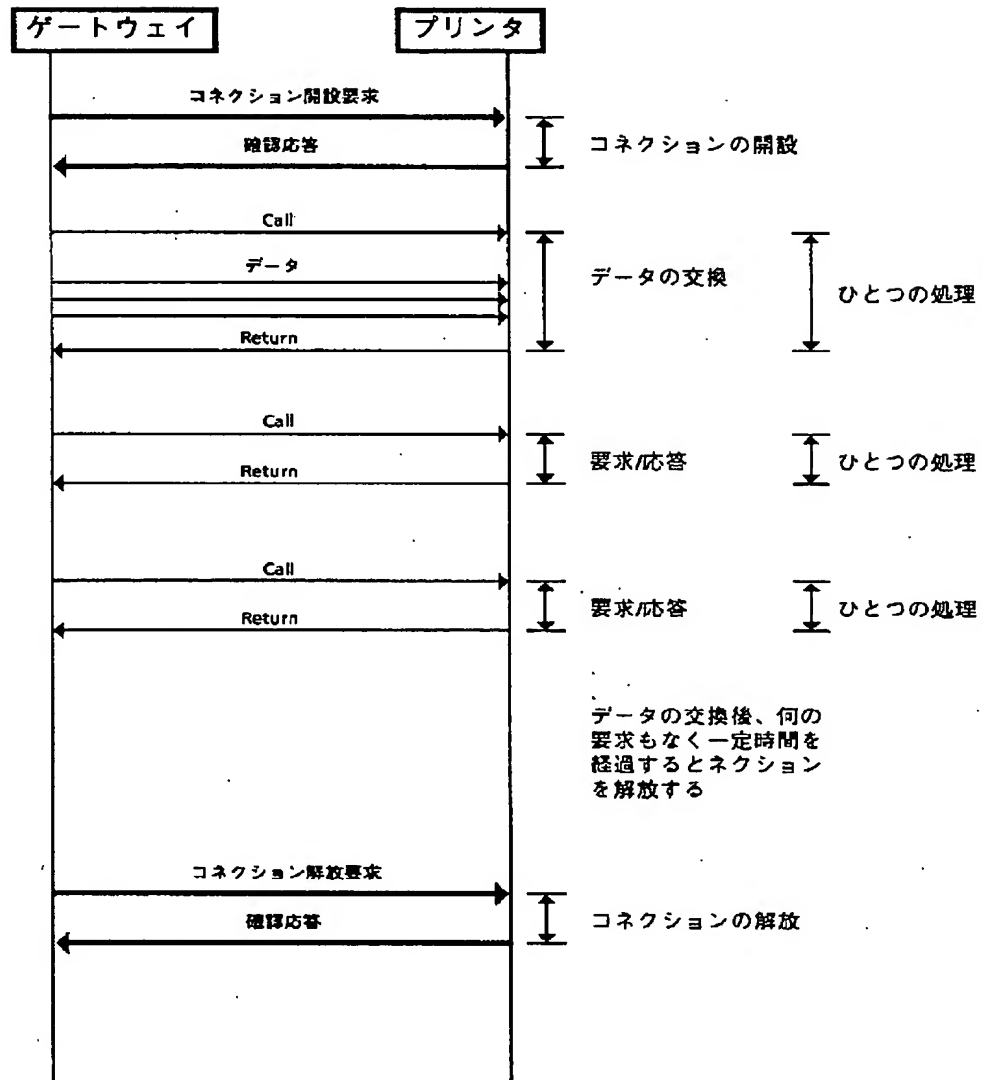
【図27】

図27 コネクションの開設・解放の処理



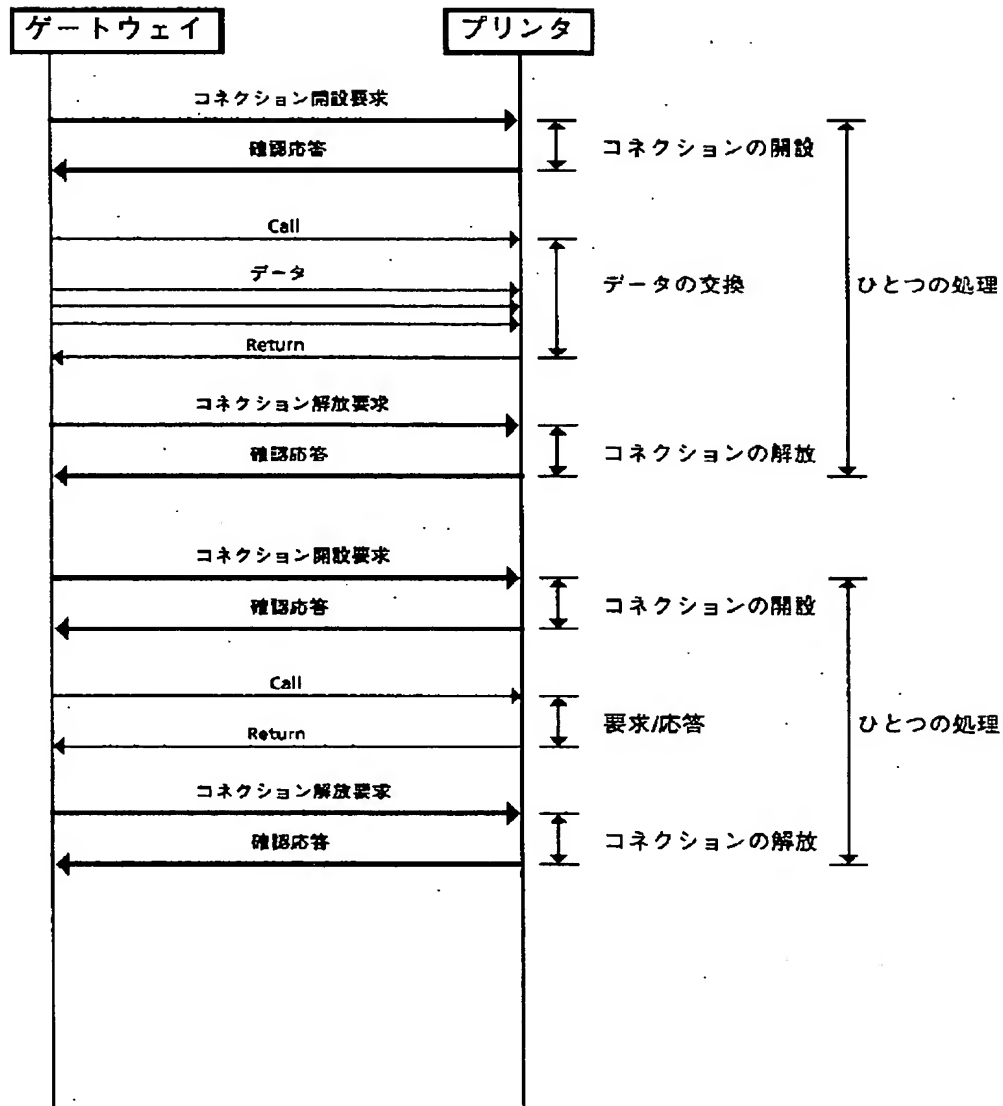
【図28】

図28 第2実施例の処理シーケンス



【図29】

図29 従来の処理シーケンス



フロントページの続き

(72)発明者 小金 武史  
 神奈川県川崎市高津区坂戸3丁目2番1号  
 KSP/R&Dビジネスパークビル 富  
 士ゼロックス株式会社内

(72)発明者 中谷 徹  
 神奈川県川崎市高津区坂戸3丁目2番1号  
 KSP/R&Dビジネスパークビル 富  
 士ゼロックス株式会社内